

web2py

ideas we stole - ideas we had

"Thanks Django, Rails, TG, Flask, CherryPy, Mako, web.py,"

by Massimo Di Pierro @ DePaul University

Multiple Users

Authentication

Authorization

Rich User Interface

Plotting

Ajax Effects



web2py

Storage

databases / caching / persistance

Connectivity

web services / JSON / XML / Mobile Devices

Main features

- One Instance - Many Applications (hot plug and play)



- Web based Integrated Development Environment
- Web based Database administration (for each app)
- Each application can connect to multiple Databases
- Writes SQL for you
- Strong on Security (no SQL Injections, XSS, CSRF, ..., audited)
- Built-in ticketing system (logs all errors)
- Runs everywhere (it is written in Python)
- Requires NO Installation (just download and unzip)
- Has no configuration files and no third party dependencies
- Can run off a USB drive
- Always backward compatible (since 2007 and on...)
- 50+ of developers already involved

Admin

The screenshot shows the Web2py Admin interface at `127.0.0.1:8000/admin/default/site`. The left sidebar lists installed applications: **admin** (currently running), **another_app**, **examples**, **friends**, and **welcome**. Each application has buttons for **edit**, **about**, **errors**, **clean**, **pack all**, **compile**, and **uninstall**.

The right panel displays system status: **Version 1.97.1 (2011-06-26 19:25:44)**, **web2py is up to date**, and **Running on Rocket 1.2.2**. It also includes links for the **New application wizard** (selected), **New simple application**, and **Upload & install packed application**. A large blue speech bubble labeled **wizard** points to the **New application wizard** link.

A blue speech bubble labeled **new app** points to the **New simple application** link. Another blue speech bubble labeled **upload app** points to the **Upload & install packed application** section. A blue speech bubble labeled **download app** points to the **admin** application entry in the sidebar. A blue speech bubble labeled **edit app** points to the **edit** button for the **admin** application.

Included APIs

- ▶ generation and parsing: HTML / XML / RSS / JSON
- ▶ web services: JSON / JSON-RPC / XML / XML-RPC / AMF
- ▶ document generation WIKI, CSV, RTF, LATEX, PDF
- ▶ 10 different SQL dialects and Google App Engine
- ▶ Role based access control with login plugins local, OpenID, OAuth 1 and 2, Janrain, LDAP Consumer + Provider Central Authentication Service
- ▶ sending SMS, accepting Credit Card payments
- ▶ internationalization, cron jobs, multi-tenancy, ...

web2py Architecture

User Applications



welcome



examples



admin

Core Libraries

HTTP request, HTTP response, session, cookies, internationalization, cache, authentication, authorization, web forms, template language, helpers, database APIs, web services APIs, etc.

rocket (ssl enabled web server)

API for third party servers (Apache,...)

python interpreter

web2py modules

```
> ls gluon/*.py
gluon/__init__.py
gluon/admin.py
gluon/cache.py
gluon/cfs.py
gluon/compileapp.py
gluon/contenttype.py
gluon/custom_import.py
gluon/dal.py
gluon/debug.py
gluon/decoder.py
gluon/fileutils.py
gluon/globals.py
gluon/highlight.py
gluon/html.py
gluon/http.py
gluon/import_all.py
gluon/languages.py
gluon/main.py
gluon/myregex.py
gluon/newcron.py
gluon/portalocker.py
```

```
gluon/reserved_sql_keywords.py
gluon/restricted.py
gluon/rewrite.py
gluon/rocket.py
gluon/sanitizer.py
gluon/serializers.py
gluon/settings.py
gluon/shell.py
gluon/sql.py
gluon/sqlhtml.py
gluon/storage.py
gluon/storage2.py
gluon/streamer.py
gluon/template.py
gluon/tools.py
gluon/utils.py
gluon/validators.py
gluon/widget.py
gluon/winservice.py
gluon/xmlrpc.py
```

web server (one file)

web2py modules

```
> ls gluon/*.py
gluon/__init__.py
gluon/admin.py
gluon/cache.py
gluon/cfs.py
gluon/compileapp.py
gluon/contenttype.py
gluon/custom_import.py
gluon/dal.py
gluon/debug.py
gluon/decoder.py
gluon/fileutils.py
gluon/globals.py
gluon/highlight.py
gluon/html.py
gluon/http.py
gluon/import_all.py
gluon/languages.py
gluon/main.py
gluon/my_regex.py
gluon/newcron.py
gluon/portalocker.py
```

```
gluon/reserved_sql_keywords.py
gluon/restricted.py
gluon/rewrite.py
gluon/rocket.py
gluon/sanitizer.py
gluon/serializers.py
gluon/settings.py
gluon/shell.py
gluon/sql.py
gluon/sqlhtml.py
gluon/storage.py
gluon/storage2.py
gluon/streamer.py
gluon/template.py
gluon/tools.py
gluon/utils.py
gluon/validator.py
gluon/winservice.py
gluon/xmlrpc.py
```

main wsgi app (one file)

web2py modules

```
> ls gluon/*.py
gluon/__init__.py
gluon/admin.py
gluon/cache.py
gluon/cfs.py
gluon/compileapp.py
gluon/contenttype.py
gluon/custom_import.py
gluon/dal.py
gluon/debug.py
gluon/decoder.py
gluon/fileutils.py
gluon/globals.py
gluon/highlight.py
gluon/html.py
gluon/http.py
gluon/import_all.py
gluon/languages.py
gluon/main.py
gluon/myregex.py
gluon/newcron.py
gluon/portalocker.py
```

```
gluon/reserved_sql_keywords.py
gluon/restricted.py
gluon/rewrite.py
gluon/rocket.py
gluon/sanitizer.py
gluon/serializers.py
gluon/settings.py
gluon/sql.py
gluon/sqlhtml.py
gluon/storage.py
gluon/storage2.py
gluon/streamer.py
gluon/template.py
gluon/tools.py
gluon/utils.py
gluon/validators.py
gluon/widget.py
gluon/winservice.py
gluon/xmlrpc.py
```

DAL / ORM (one file)

web2py modules

```
> ls gluon/*.py
gluon/__init__.py
gluon/admin.py
gluon/cache.py
gluon/cfs.py
gluon/compileapp.py
gluon/contenttype.py
gluon/custom_import.py
gluon/dal.py
gluon/debug.py
gluon/decoder.py
gluon/fileutils.py
gluon/globals.py
gluon/highlight.py
gluon/html.py
gluon/http.py
gluon/import_all.py
gluon/languages.py
gluon/main.py
gluon/myregex.py
gluon/newcron.py
gluon/portalocker.py
```

```
gluon/reserved_sql_keywords.py
gluon/restricted.py
gluon/rewrite.py
gluon/rocket.py
gluon/sanitizer.py
gluon/serializers.py
gluon/settings.py
gluon/shell.py
gluon/sql.py
gluon/sqlhtml.py
gluon/storage.py
gluon/storage2.py
gluon/streamer.py
gluon/template.py
gluon/tools.py
gluon/utils.py
gluon/validators.py
gluon/widget.py
gluon/winservice.py
gluon/xmlrpc.py
```

template (one file)

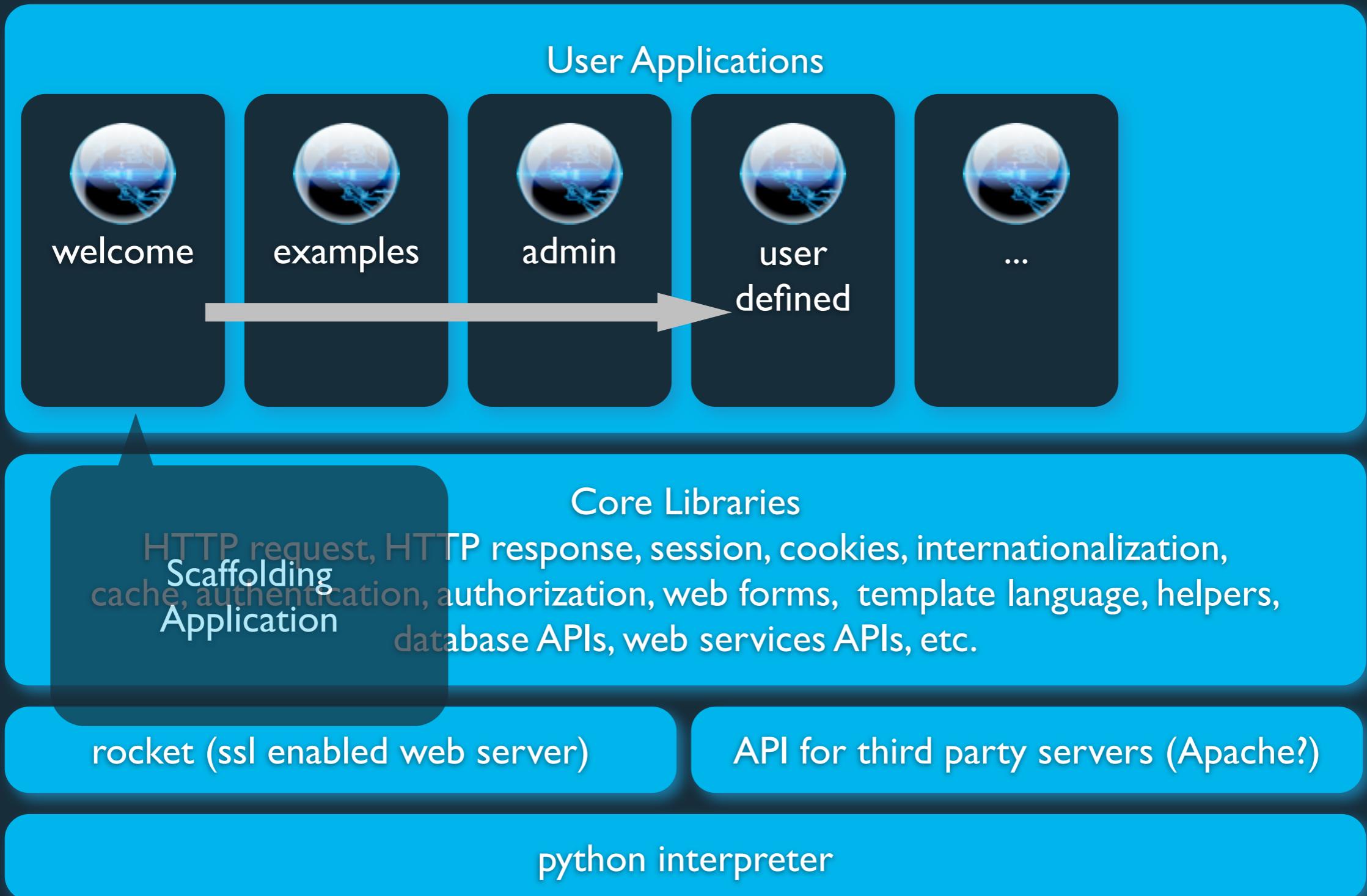
web2py modules

```
> ls gluon/*.py
gluon/__init__.py
gluon/admin.py
gluon/cache.py
gluon/cfs.py
gluon/compileapp.py
gluon/contenttype.py
gluon/custom_import.py
gluon/dal.py
gluon/debug.py
gluon/decoder.py
gluon/fileutils.py
gluon/globals.py
gluon/highlight.py
gluon/html.py
gluon/http.py
gluon/import_all.py
gluon/languages.py
gluon/main.py
gluon/myregex.py
gluon/newcron.py
gluon/portalocker.py
```

```
gluon/reserved_sql_keywords.py
gluon/restricted.py
gluon/rewrite.py
gluon/rocket.py
gluon/sanitizer.py
gluon/serializers.py
gluon/settings.py
gluon/shell.py
gluon/sql.py
gluon/sqlhtml.py
gluon/storage.py
gluon/storage2.py
gluon/streamer.py
gluon/timer.py
gluon/tools.py
gluon/utils.py
gluon/validators.py
gluon/widget.py
gluon/winservice.py
gluon/xmlrpc.py
```

helpers (one file)

web2py Architecture



web2py Architecture



User Applications



welcome



examples



admin



user
defined



...



uploaded

Core Libraries

HTTP request, HTTP response, session, cookies, internationalization, cache, authentication, authorization, web forms, templates, database APIs, web services APIs, etc.

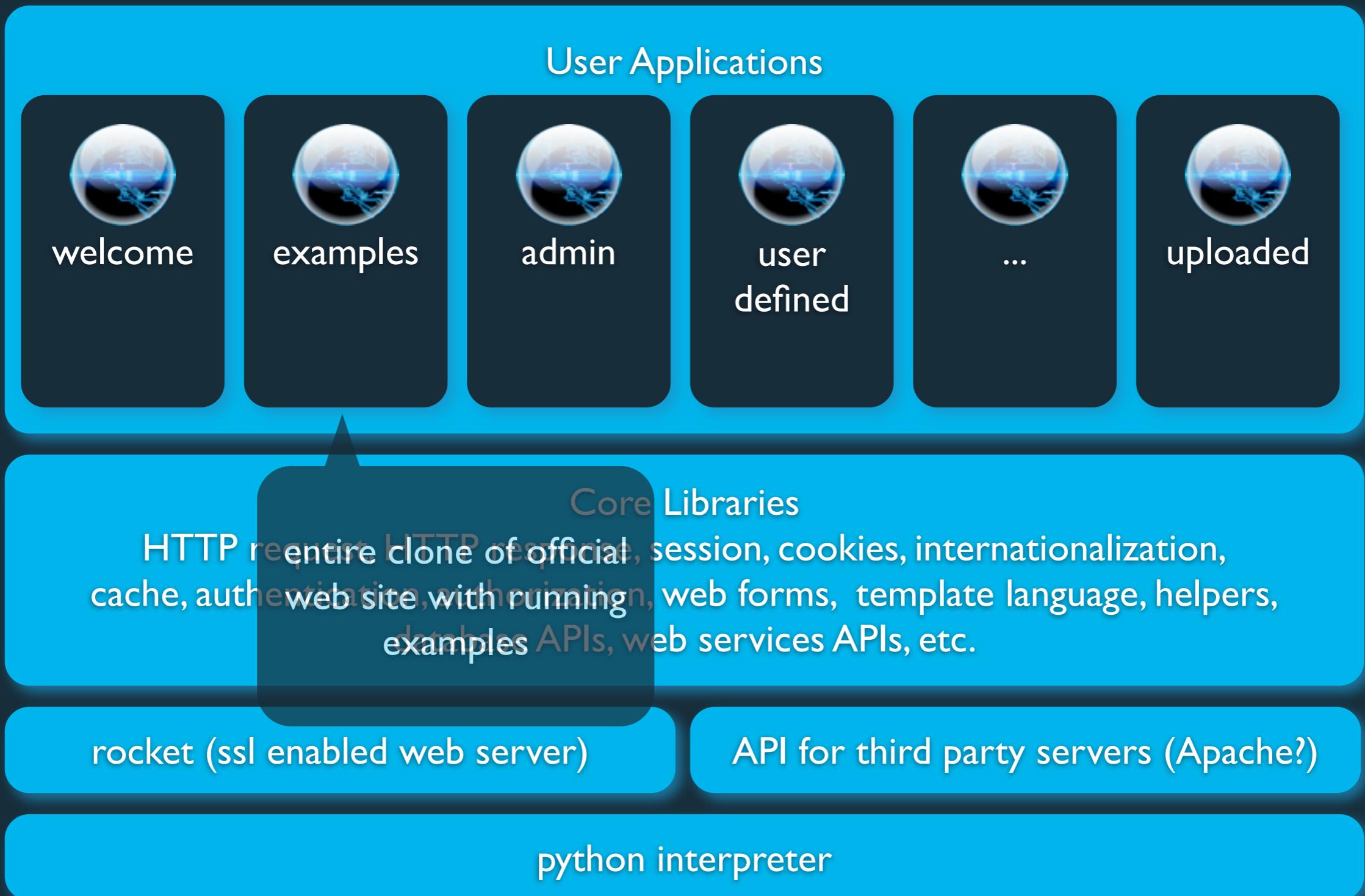
Applications can be downloaded and installed remotely

rocket (ssl enabled web server)

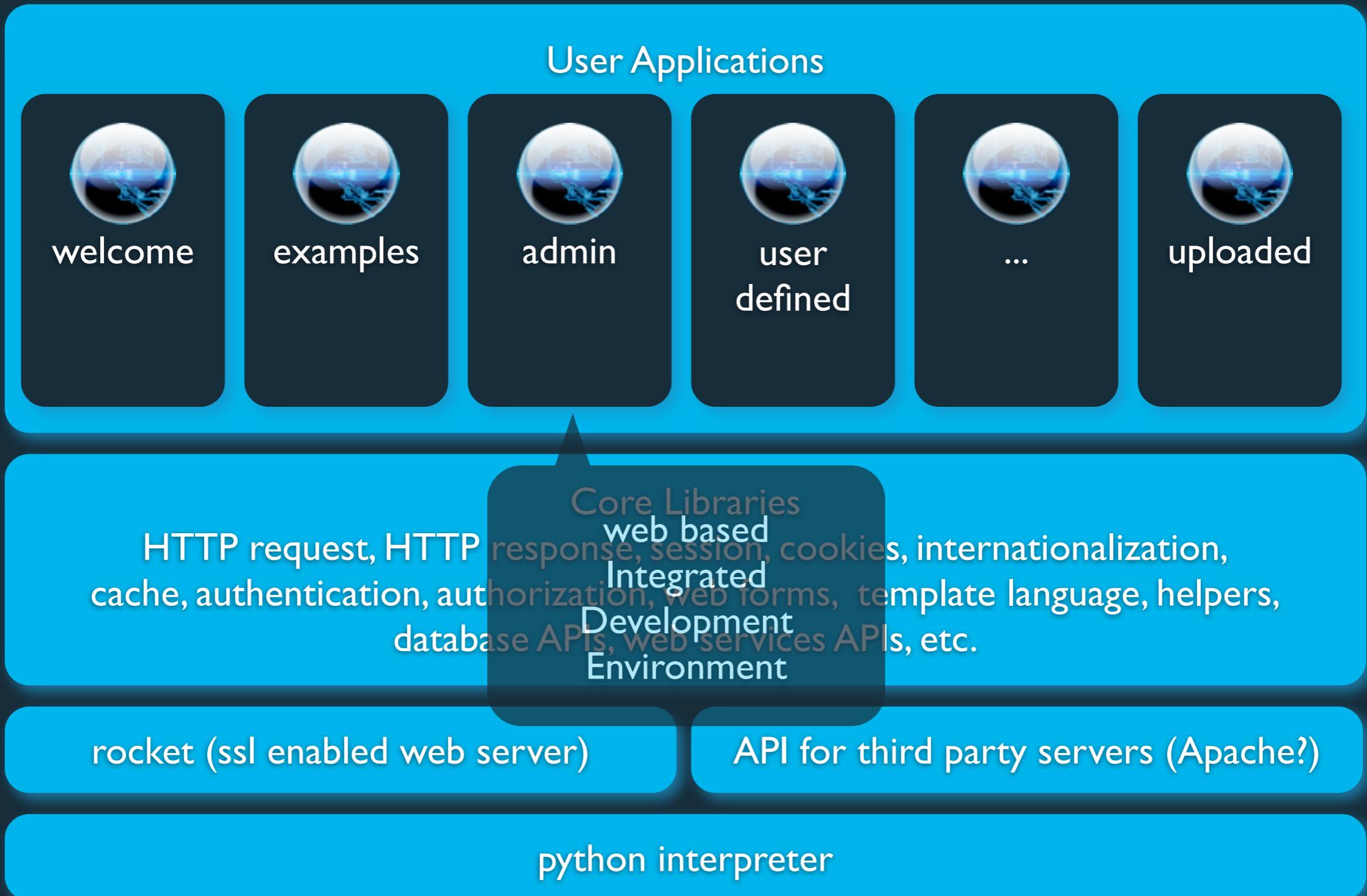
API for third party servers (Apache?)

python interpreter

web2py Architecture

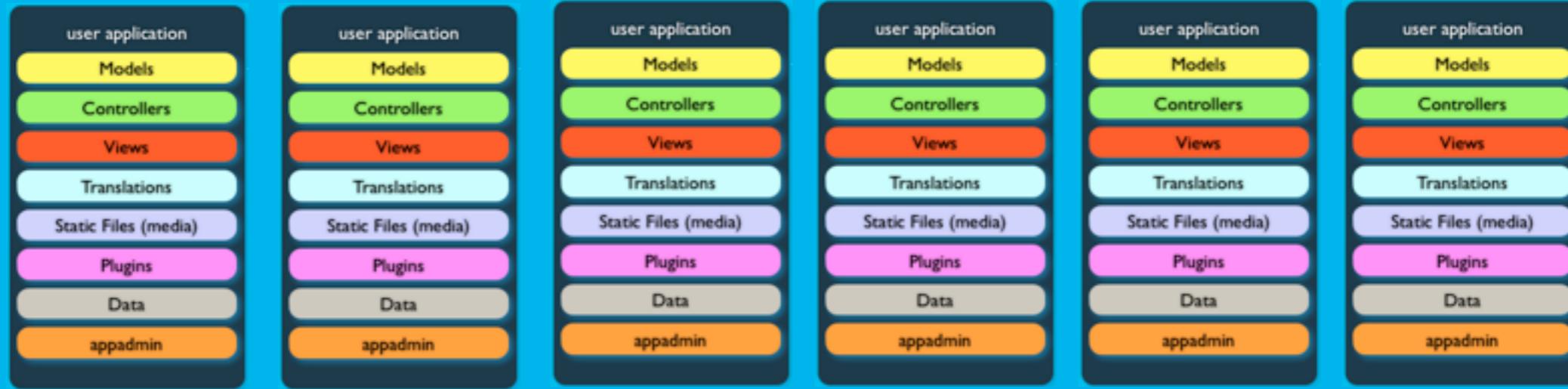


web2py Architecture



web2py Architecture

User Applications



Core Libraries

HTTP request, HTTP response, session, cookies, internationalization, cache, authentication, authorization, web forms, template language, helpers, database APIs, web services APIs, etc.

rocket (ssl enabled web server)

API for third party servers (Apache?)

python interpreter

Admin - Design

design friends

127.0.0.1:8000/admin/default/design/friends

WEB2PY

site edit about errors versioning logout help

EDIT APPLICATION "FRIENDS"

Models

Controllers

shell test crontab

edit x appadmin.py exposes index, insert, download, csv, select, update, state, ccache
edit x default.py exposes index, user, download, call, data
edit x main.py exposes index

create file with filename: create

Views

Languages

Static files

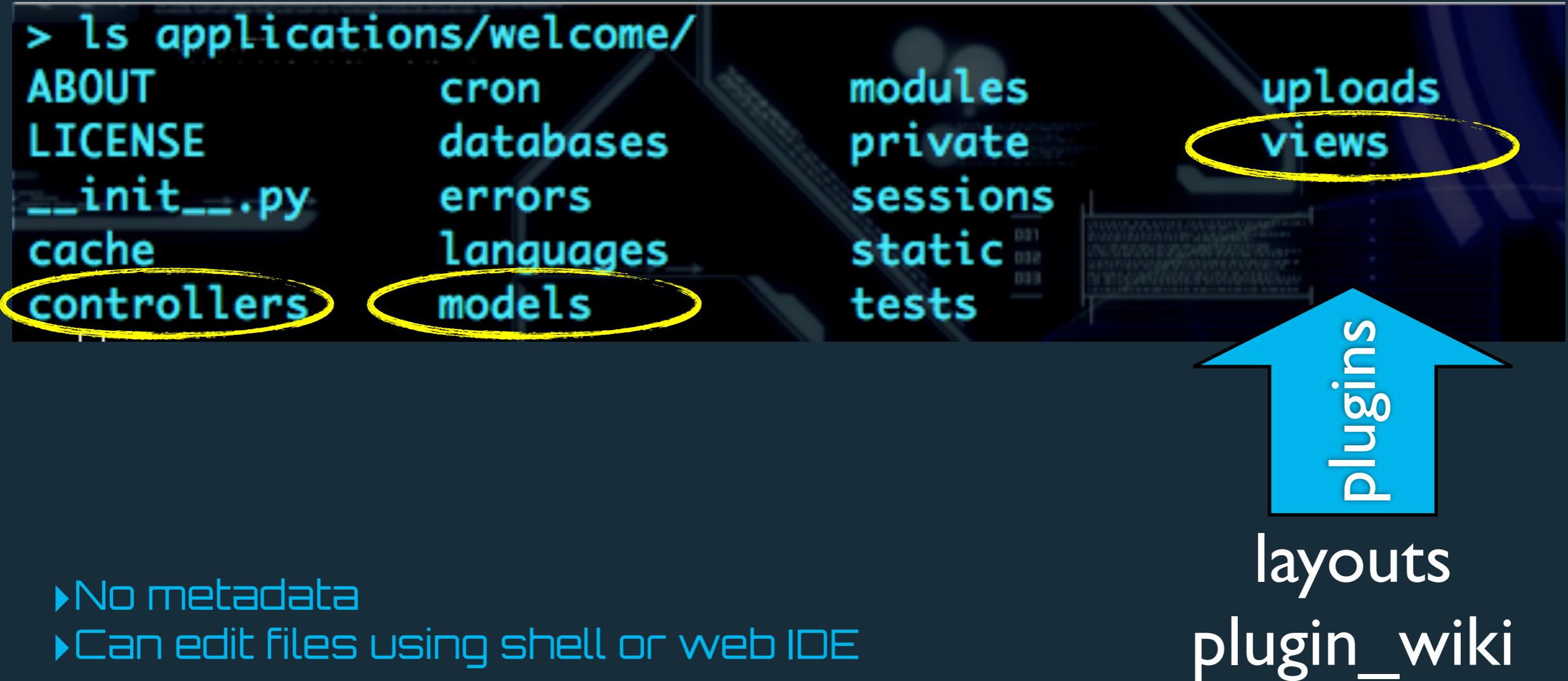
Modules

Plugins

Powered by web2py™ created by Massimo Di Pierro ©2007-2010 - Admin language en

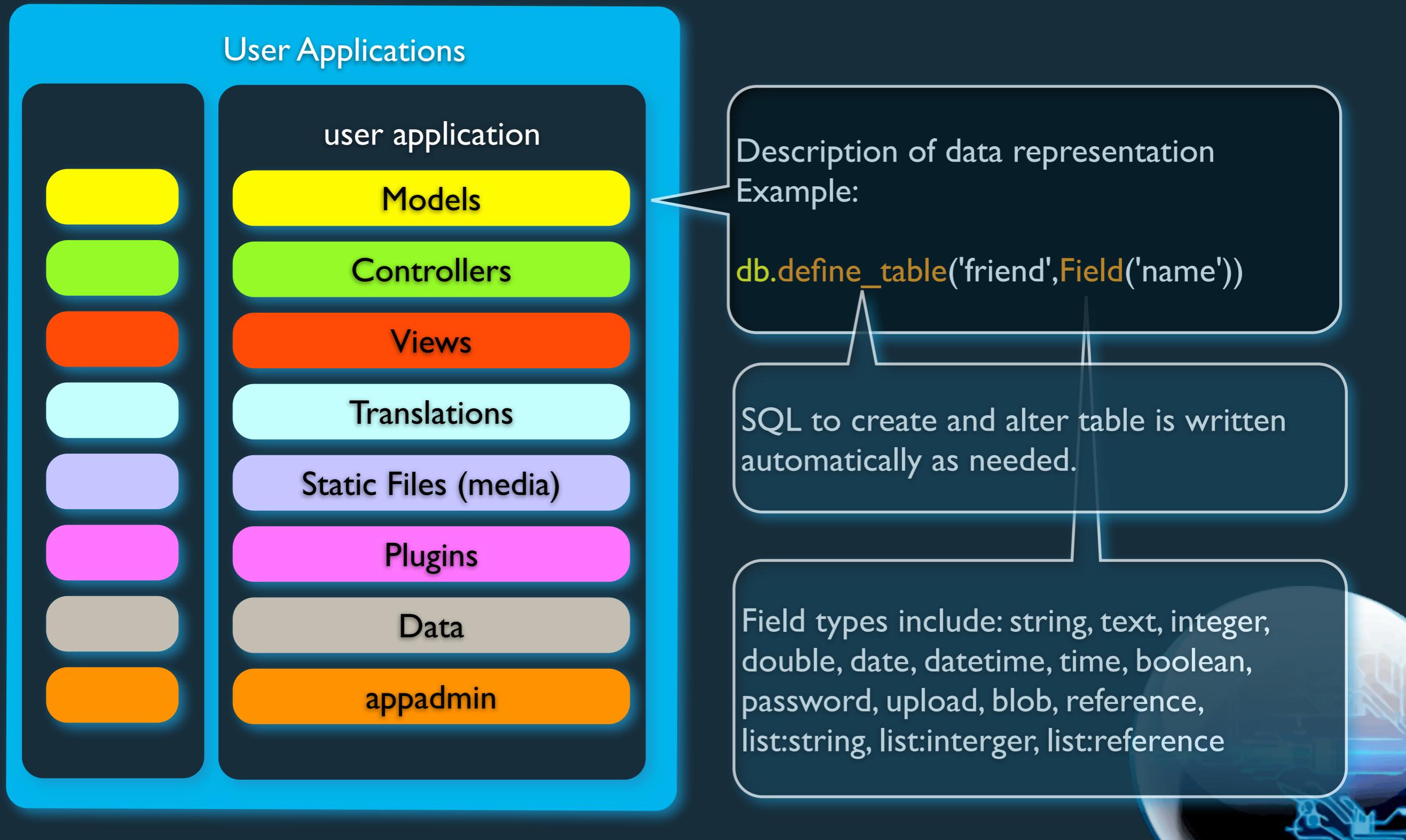
plugin_wiki (CMS)

web2py applications

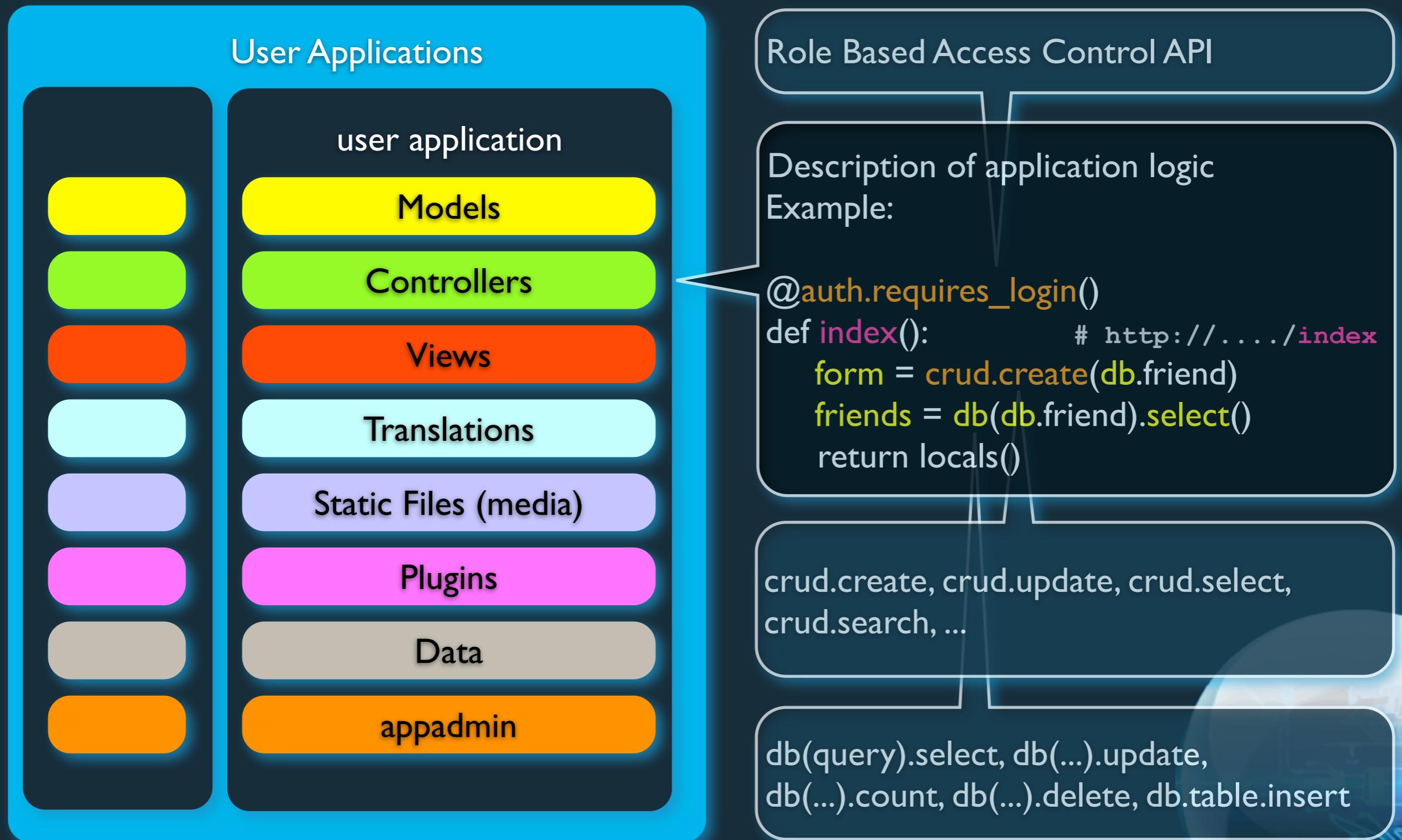


- ▶ No metadata
- ▶ Can edit files using shell or web IDE

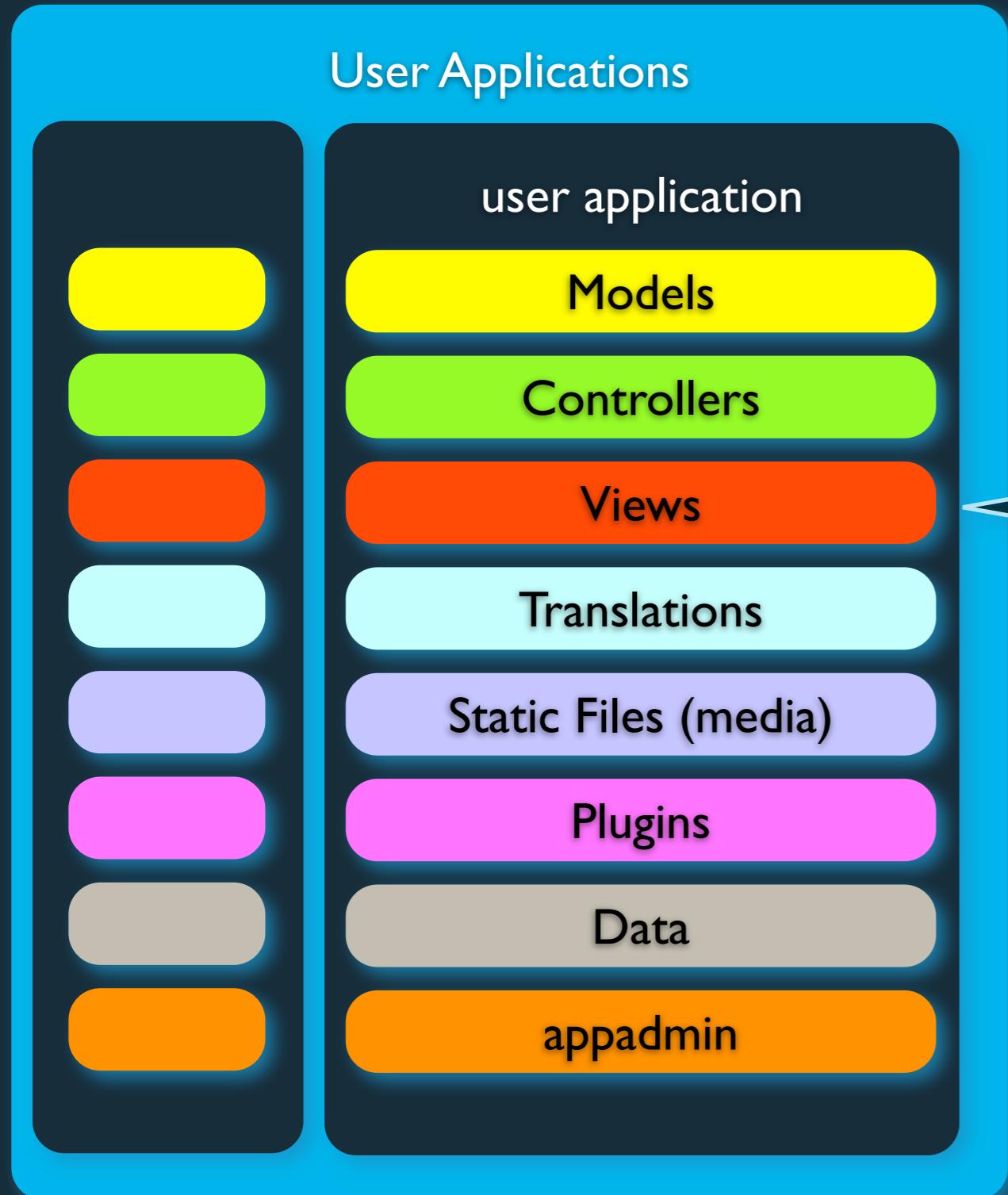
Architecture of Applications



Architecture of Applications



Architecture of Applications

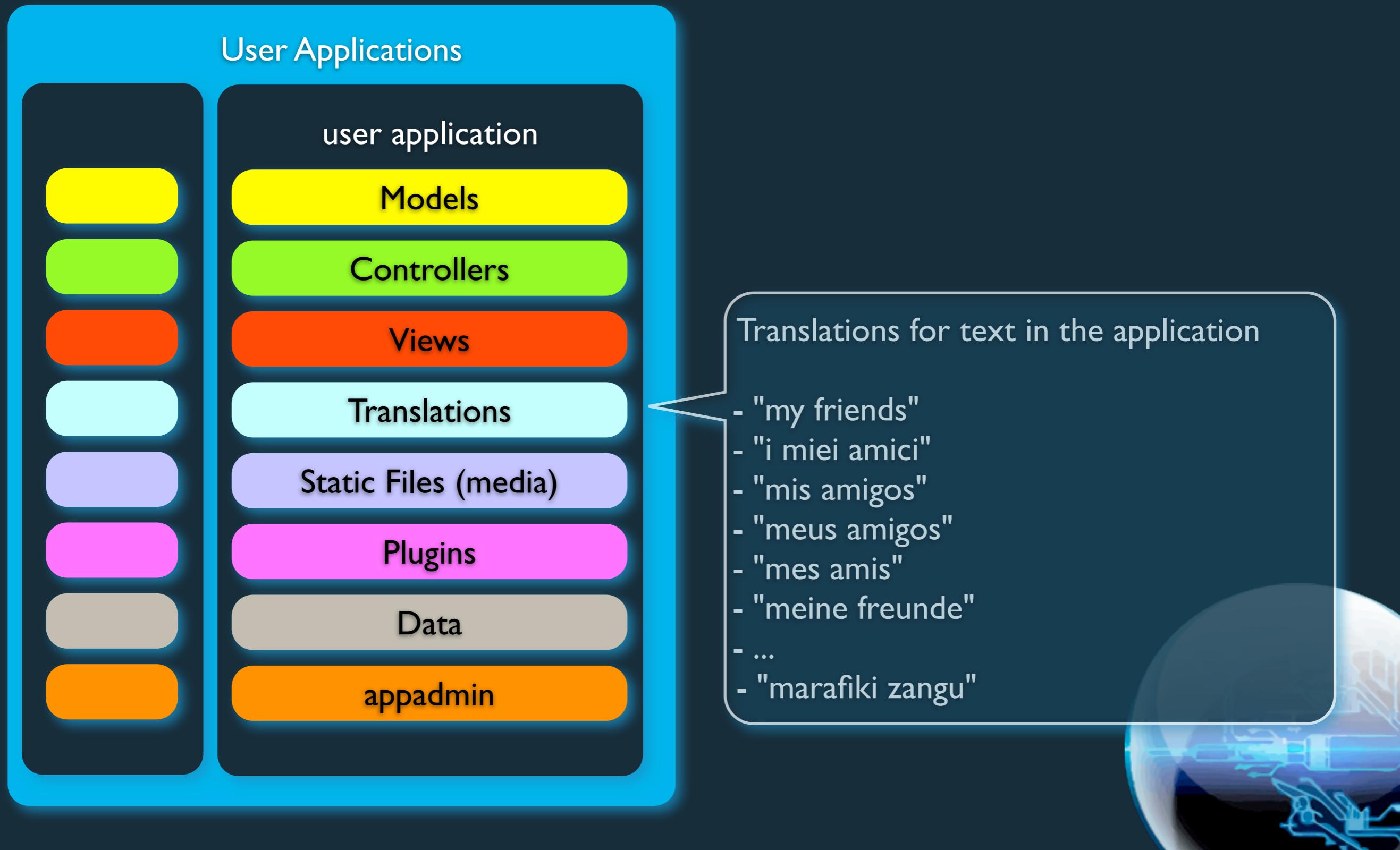


Description of data presentation
Example:

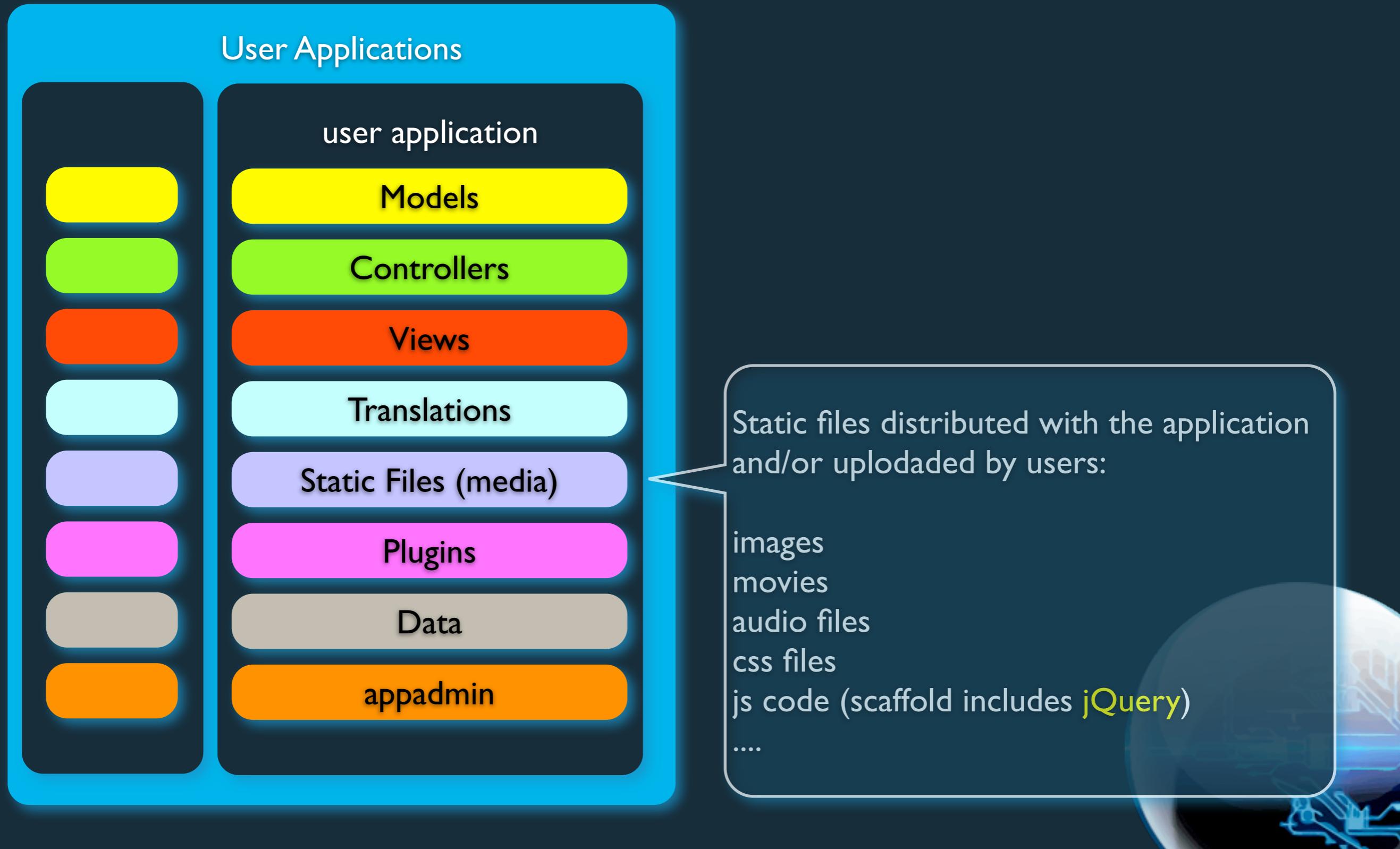
```
 {{extend 'layout.html'}}  
<h1>{{=T('My Friends')}}</h1>  
<h2>New Friend</h2>  
 {{=form}}  
<h2>Current Friends</h2>  
 {{=friends}}
```

embeds and renders any object in the page
full python allowed in {{...python...}}
including loops and function definitions.

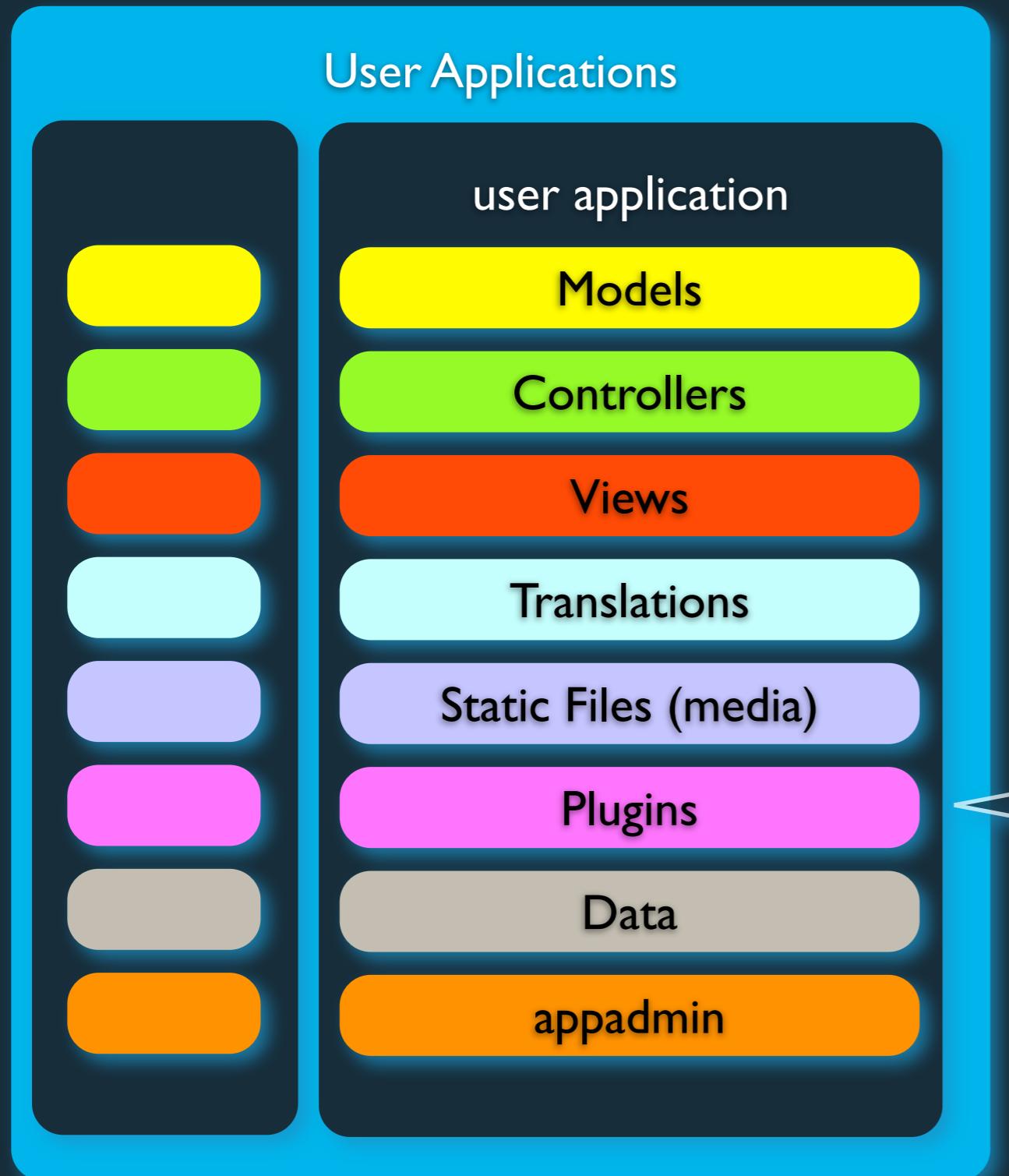
Architecture of Applications



Architecture of Applications



Architecture of Applications

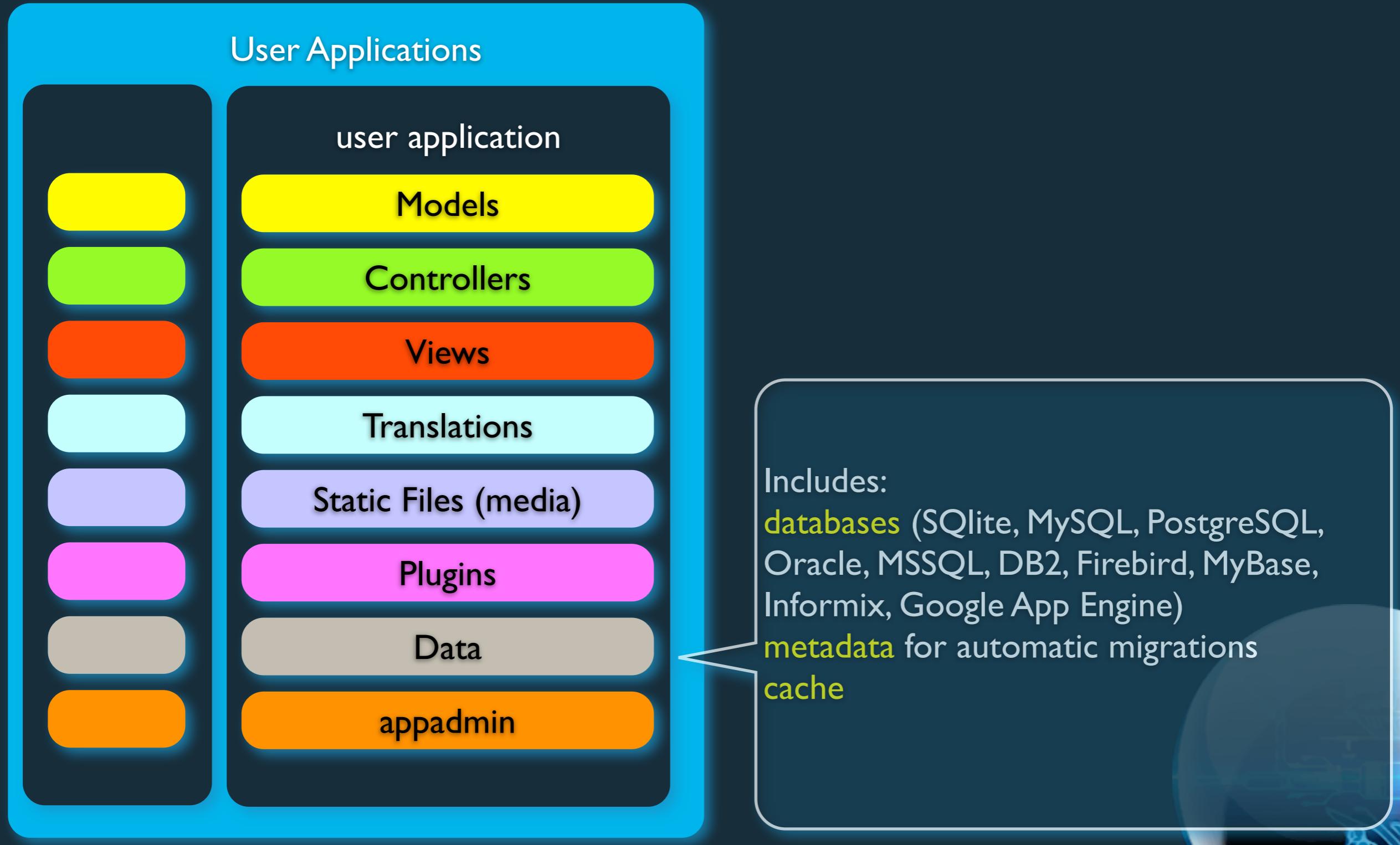


Any subset of an application can be packaged and can be distributed. This is called a plugin. Often plugins define components, i.e. functional elements that can be embedded in pages.

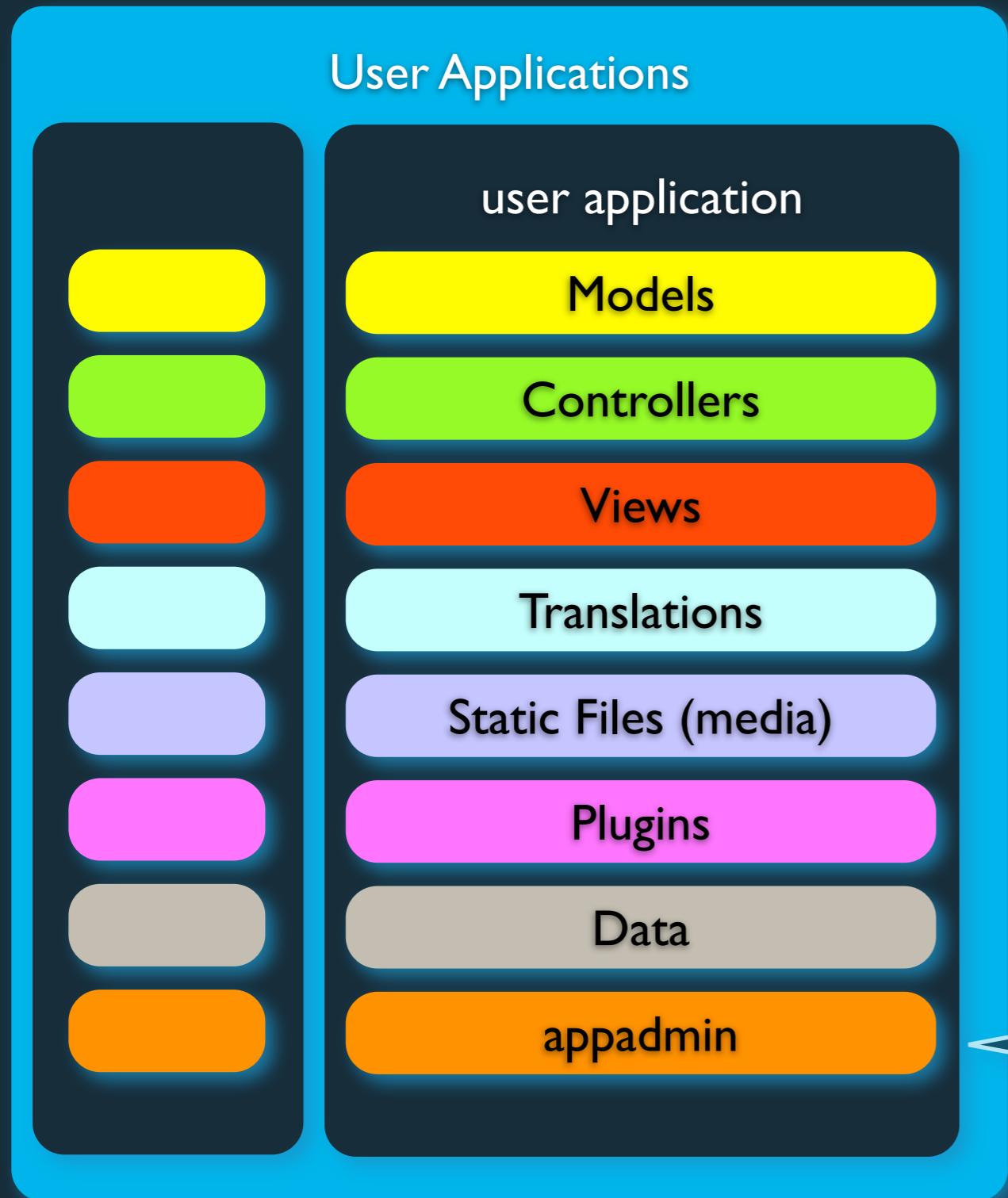
Example:

plugin_wiki adds a CMS to your app
plugin_mobile makes it iphone look-alike

Architecture of Applications



Architecture of Applications



default web based interface to your data

Complete Application ("friends")

File: "friends/models/db_friend.py"

```
db.define_table('friend',Field('name'))
```

File: "friends/controllers/main.py"

```
@auth.requires_login()
def index():
    form = crud.create(db.friend)
    friends = db(db.friend).select()
    return locals()
```

File: "friends/views/main/index.html"

```
{{extend 'layout.html'}}
<h1>{{=T('My Friends')}}</h1>
<h2>New Friend</h2>
{{=form}}
<h2>Current Friends</h2>
{{=friends}}
```

web2py/
applications/
friends/
models/db_friends.py
controllers/main.py
views/main/index.html
...

FriendsApp

Home This App Resources

My Friends

New Friend

Name:

Submit

Current Friends

friend.id	friend.name
1	test
2	Max
3	John



```

cd /path/to/demo
wget -O web2py_src.zip http://web2py.com/examples/static/web2py_src.zip
unzip -o -q web2py_src.zip
cd web2py
python web2py.py -a hello -p 8000 &
cd applications
mkdir friends
cp -r welcome/* friends/
cd friends

echo "db.define_table('friend',Field('name'))" > models/db_friends.py

echo "
@auth.requires_login()
def index():
    form = crud.create(db.friend)
    friends = db(db.friend).select()
    return locals()
" > controllers/main.py

mkdir views/main
echo "
{{extend 'layout.html'}}
<h1>{{=T('My Friends')}}</h1>
<h2>New Friend</h2>
{{=form}}
<h2>Current Friends</h2>
{{=friends}}
" > views/main/index.html

```

FriendsApp

Home This App Resources

My Friends

New Friend

Name:

Current Friends

friend.id	friend.name
1	test
2	Max
3	John

Controllers

Django (view in MTV)

```
def index(request):
    entry_id = request.GET['entry_id']
    entry = Entry.objects.get(pk=entry_id)
    output = entry.name
    return HttpResponseRedirect(output)
```

STOLEN

web2py (controller in MVC)

```
def index():
    entry_id = request.get_vars.entry_id or redirect(URL('error'))
    entry = Entry(entry_id)
    output = entry.name
    return dict(output=output) # defaults to generic template
```

Routing (in, out, onerror)

Django (urls.py)

```
urlpatterns = patterns('',
    (r'^articles/$', 'news.views.index'),
    (r'^articles/(\d{4})/$', 'news.views.read'),
)
```

STOLEN

web2py (routes.py) - ALWAYS OPTIONAL

```
routes_in = [
    (r'articles/', '/news/default/index'),
    (r'articles/(\d{4})', '/news/default/read/\1'),
    (r'articles/$year', '/news/default/read/$year'),
    (r'127.0.0.*:http://domain.com articles/(\d{4})', '/news2/default/read/\1')
]

routes_out = [...]

routes_onerror = [
    (r'init/400', r'/init/default/login'),
    (r'/*/*', r'/init/static/fail.html')]
```

Templates

Mako (template in MTV)

```
<%inherit file="base.html"/>

<%def name="makerow(k)">
  <tr>
    <td>${k}</td>
    <td>${k*k}</td>
  </tr>
</%def>

<%
  numbers = range(0,10)
%>





```

STOLEN

web2py (template or view in MVC)

```
{{extend "base.html"}}

{{ def makerow(k): }}
  <tr>
    <td>{{=k}}</td>
    <td>{{=k*k}}</td>
  </tr>
{{ return }}

{{ 
  numbers = range(0,10)
}}





```

- ▶ In web2py, no indentation requirement
- ▶ Django-like blocks, output always escaped

App-Admin

Django

Django administration

Welcome, jacob. Documentation / Change password / Log out

Site administration

Auth	Add Change
Groups	Add Change
Users	Add Change
Sites	Add Change
CMS	
Authors	Add Change
Books	Add Change
Publishers	Add Change

Recent Actions

My Actions
None available

web2py

A0App

[design](#) [db](#) [state](#) [cache](#)

Available databases and tables

[db.auth_user](#)

[\[insert new auth_user\]](#)

[db.auth_group](#)

[\[insert new auth_group\]](#)

[db.auth_membership](#)

[\[insert new auth_membership\]](#)

[db.auth_permission](#)

[\[insert new auth_permission\]](#)

[db.auth_event](#)

[\[insert new auth_event\]](#)

[db.auth_cas](#)

[\[insert new auth_cas\]](#)

[\[login\]](#) [\[register\]](#) [\[lost password?\]](#)

database db select

[\[insert new auth_user\]](#)

Rows in table

Query:

Update:

Delete:

The "query" is a condition like "db.table1.field1=='value'". Something like "db.table1.field1==db.table2.field2" results in a JOIN.

Use (...&...) for AND, (...|...) for OR, and <...> for NOT to build more complex queries.

"update" is an optional expression like "field1=newvalue". You cannot update or delete the results of a JOIN.

1 selected

auth_user.id auth_user.first_name auth_user.last_name auth_user.email

1	Massimo	Di Piero	mpierro@is...	8074502	2009-07-01 11:11:07

Import/Export

[\[export as csv file\]](#)

or import from csv file No file chosen



- ▶ Django "admin" designed for public access
- ▶ web2py "app-admin" designed for administrator access only
- ▶ CRUD components from appadmin can be embedded in apps
- ▶ web2py "app-admin" not to be confused with web2py's "admin"

Models

Django (model)

```
class Entry(models.Model):
    name      = models.CharField(max_length=255,null=False)
    body      = models.TextField()
    image     = models.ImageField()
    pub_date  = models.DateTimeField()
    rating    = models.IntegerField()
```

STOLEN

web2py (model)

```
Entry = db.define_table('entry',
    Field('name',length=255,notnull=True),
    Field('body','text'),
    Field('image','upload',requires=IS_IMAGE()),
    Field('pub_date','datetime'),
    Field('rating','integer')]
```

Queries

Django (model)

```
q = Entry.objects.filter(headline__startswith="What")
q = q.filter(pub_date__lte=datetime.now())
q = q.exclude(body__icontains="food")
print q
```

web2py (model)

```
q = Entry.headline.startswith("What")
q = q & (Entry.pub_date<datetime.now())
q = q & (!Entry.body.contains("food"))
print db(q).select()
```

Thread Locals

STOLEN

Flask (proxies to objects that are local to a specific context)

```
from flask import request

with app.request_context(environ):
    assert request.method == 'POST'
```

web2py (thread-locals)

```
from gluon import current

assert current.request.env.http_method == 'POST'
```

- ▶ In Flask "request" is a proxy to a thread local object. In web2py "request" is a thread local object contained into a "current"
- ▶ "gluon" is the library that contains web2py.

Multi-version / No-conflicts

<https://github.com/mitsuhiko/multiversion>

```
import multiversion
multiversion.require_version('mylib', '1.0')
import mylib
```

web2py (each app can ship with its own version of libraries)

```
# app 1
import mylib      # from applications/app1/modules/

# app 2
import mylib      # from applications/app2/modules/
```

- ▶ Each app ships with its own modules/ folder.
- ▶ Not added to sys.path
- ▶ No conflicts
- ▶ One web2py instance

Role based Access Control



For any function

```
@auth.requires_login()  
@auth.requires_membership(role='secret agent')  
@auth.requires_permission('kill', 'bad_people', all)  
def test():  
    return 'done'
```

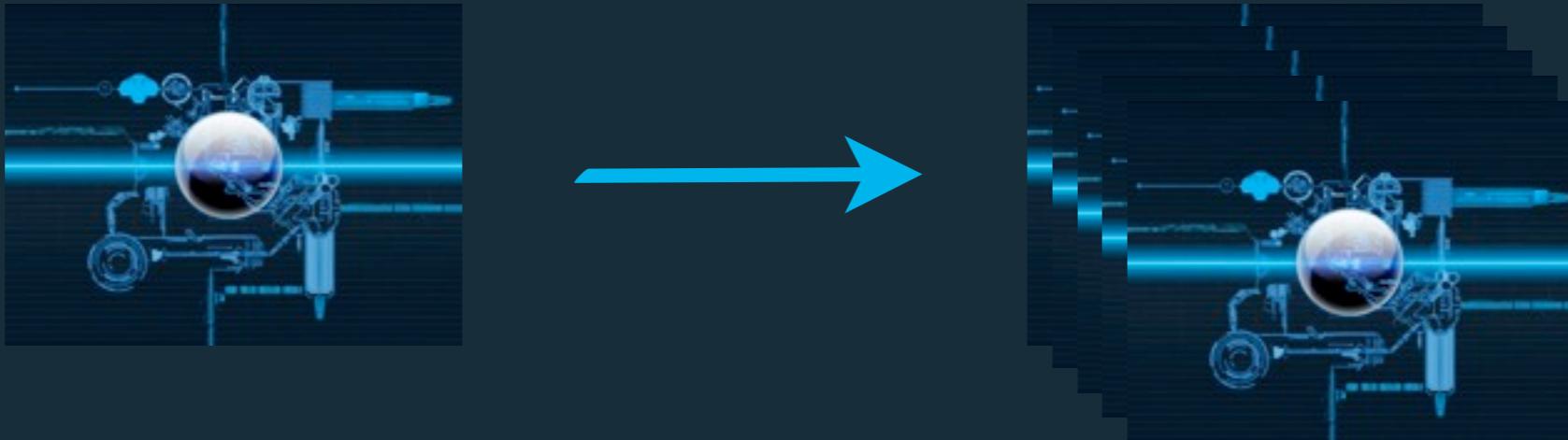
Web Services



For any function

```
@service.json  
@service.xml  
@service.jsonrpc  
@service.xmlrpc  
@service.soap  
@service.amfrpc3('domain')  
def add(a,b):  
    return a+b
```

Record Versioning

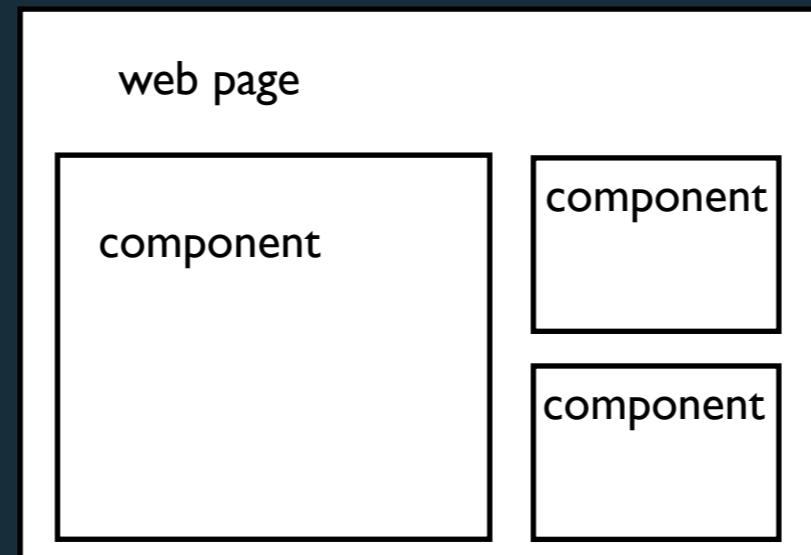


Any method

```
db._common_fields.append(auth.signature)  
crud.settings.update_onaccept(crud.archive)
```

- ▶ Store all previous version of each record with names of the user who changed and timestamp of the change

Modularity with Digitally Signed URLs



In page

```
{=LOAD('plugin','component',user_signature=True)}
```

In controller

```
@auth.requires_signature()  
def component():  
    return 'component'
```

- ▶ web2py can sign all URLs (links, ajax callbacks, components) so the called action delegates security the caller

Federated Authentication



App "one"

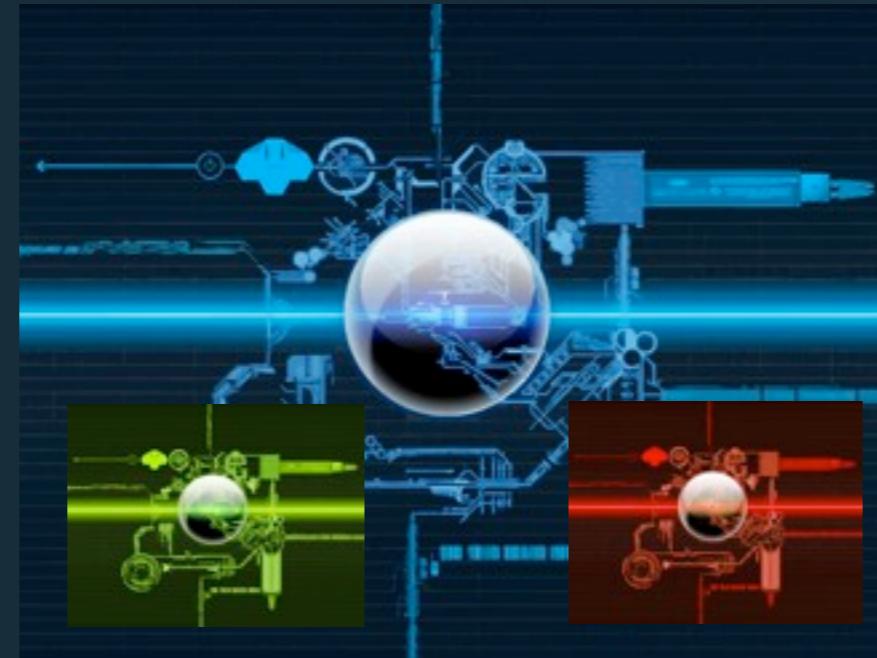
```
auth = Auth(db)
```

Other apps

```
auth = Auth(db, cas_provider = 'http://.../one/default/user/cas')
```

- ▶ Any application can be both a provider and a client for CAS 2.0
- ▶ Other federated authentication mechanism available as clients

Multi-tenancy



Any model

```
db._common_fields.append(Field('request_tenant', default=request.env.host_name))
```

- ▶ All records are filtered based on tenant ownership
- ▶ Tenant identified for example by domain name
- ▶ Tables can be shared between tenant or not
- ▶ <http://domain1> or <http://domain2> (same app, different data)

GAE Deployment

WEB2PY

● GOOGLE APP ENGINE DEPLOYMENT INTERFACE

This page can upload your application to the Google App Engine computing cloud. Mind that you must first create this is done by installing the Google appserver and running the app locally with it once, or there will be errors when Attention: deployment may take long time, depending on the network speed. Attention: it will overwrite your app TWICE.

● Deployment form

Path to appcfg.py:

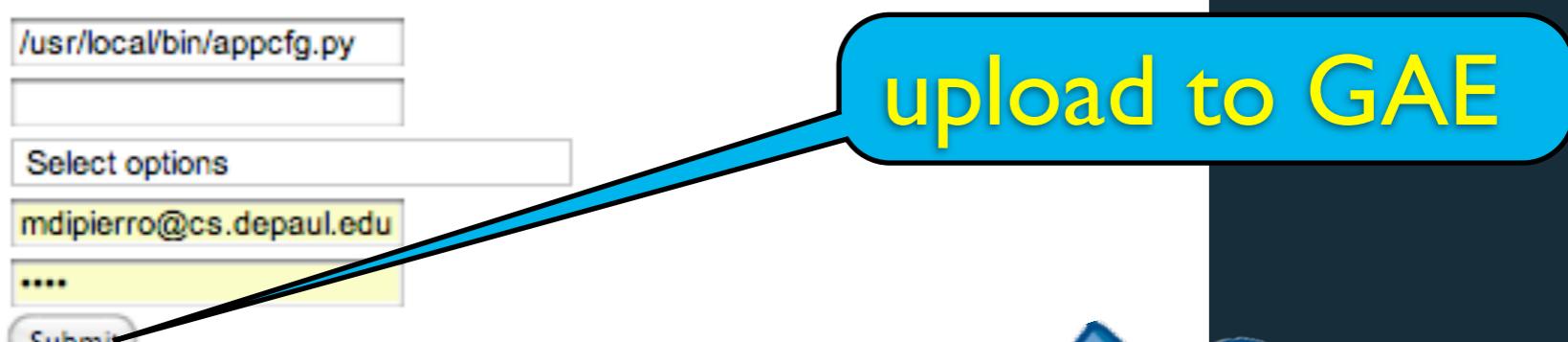
Google Application Id:

web2py apps to deploy:

GAE Email:

GAE Password:

upload to GAE



Web translation

WEB2PY

EDITING LANGUAGE FILE "WELCOME/LANGUAGES/
ORIGINAL/TRANSLATION

"update" is an optional expression like "field1='newvalue'". You cannot update or delete the results of a JOIN query directly. "update" è un'espressione opzionale come "campo1='nuovo valore'". Non si può fare "update" o "delete" dei risultati di un JOIN

%s rows deleted
%s righe ("record") cancellate

%s rows updated
%s righe ("record") modificate

%Y-%m-%d
%d/%m/%Y

%Y-%m-%d %H:%M:%S
%d/%m/%Y %H:%M:%S

Administrative interface
Interfaccia amministrativa

english

italian

The diagram illustrates the mapping between English and Italian language entries. It consists of two columns of text pairs, each with a 'delete' button. Blue arrows point from the English labels ('english') to their corresponding Italian translations ('italian').

- english: %s rows deleted → italiano: %s righe ("record") cancellate
- english: %s rows updated → italiano: %s righe ("record") modificate
- english: %Y-%m-%d → italiano: %d/%m/%Y
- english: %Y-%m-%d %H:%M:%S → italiano: %d/%m/%Y %H:%M:%S
- english: Administrative interface → italiano: Interfaccia amministrativa

Error logging

The screenshot shows a Web2py application interface with two main sections: 'ERROR LOGS FOR "A1"' and 'ERROR TICKET FOR "A1"'.

ERROR LOGS FOR "A1":

- Header: WEB2PY, Home icon.
- Section title: ● ERROR LOGS FOR "A1"
- Buttons: check all, uncheck all, delete all checked.
- Text: Click row to expand traceback.
- Link: lists by ticket.
- Table header: Delete Count File Error.
- Data rows:
 - 3 default.py ZeroDivisionError: integer division or modulo by zero + details
 - 1 default.py NameError: name 'asasdf' is not defined + details

error occurred 3 times (highlighted by a blue callout pointing to the first log entry)

ERROR TICKET FOR "A1":

- Header: WEB2PY, Home icon.
- Section title: ● ERROR TICKET FOR "A1"
- Text: d06-fbab-4518-beeb-8f3cda692432
- Text: 04 09:10:10)
- Text: frameworks/Python.framework/Versions/2.5/Resources/Python.app/
- Text: oops: I/O
- Text: 1. Traceback (most recent call last):
2. File "/Users/mdipierro/web2py/agnon/restricted.py", line 192, in restricted
3. exec code in environment
4. File "/Users/mdipierro/web2py/applications/ag1/controllers/default.py", line 12, in <module>
5. 1/0
6. ZeroDivisionError: integer division or modulo by zero
7.

oops: I/O (highlighted by a blue callout pointing to the error message in the ticket details)

Who uses web2py?



Conclusions

- ▶ web2py has been abround for since 2007
- ▶ +50% was rewritten in 2010 while mantaining backward compatibility
- ▶ Some like it, some find it useful
- ▶ Give it a try!



@zedshaw
zedshaw

I am dead sick of the Python community's sense of "purity" bullshit making the language and platform harder to use. Magic rocks, I want more