



# Poisson Magique project

et introduction à le cadre de logiciel "Lamson"

Pablo Duboue

<http://poissonmagique.net>

# Poisson Magique 30 sec demo



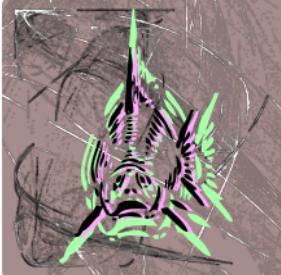
.poissonmagique.net    [poissonmagique.net](#)    [Source Code](#)

[Log in](#)    or    [Sign up](#)

## Welcome to Poisson Magique!

Play-by-email role-playing game gateway and Web interface with an emphasis in Call of Cthulhu.

[Learn more »](#)



### For the GM

Poisson Magique is a mail server + web interface to help with the GM tasks of modifying and forwarding emails in a PbP campaign.

Each of the emails sent by the users can be split, re-assembled and modified for forwarding.

[View details »](#)

### For the Player

Each character (player and non-player) has an unique email address on the server (e.g., john.smith@...). The player can communicate seamlessly with each character by emailing them (the email might get refused if the player is not in the same room).

[View details »](#)

Poisson Magique, Copyright (C) 2013 by [Pablo Duboue](#).  
This software comes with NO WARRANTY and it is available under the terms of the [AGPL-3.0 license](#).

# Poisson Magique 30 sec demo

```
@route("gm@(host)")  
@bounce_to(soft=GM_BOUNCE, hard=GM_BOUNCE)  
def START(message, host=None):  
    human = _check_sender(message)  
    if human is None:  
        return  
    campaign = find_campaign_for_sender(human)  
    if campaign is None:  
        return NO_GAME(message)  
    gm = campaign.gm  
    if gm != human and not gm.is_bouncing:  
        c = find_character(human)  
        new_from = "%s <%s>" % (c.name, c.mail_address)  
  
        new_message = MailResponse(To=gm.mail_address,  
                                  From=new_from,  
                                  Subject=message['Subject'],  
                                  Body=message.body()))  
        new_message.attach_all_parts(message)  
  
        relay.deliver(new_message)
```

# Jeu de rôle (activité ludique)



- Le jeux de rôle sur table sont considères comme une forme de narration (storytelling) interactive et collaborative :
  - la structure des événements, des personnages et de la narration offrent une expérience narrative, sans nécessiter un scénario fortement défini.
- L'interactivité est la différence essentielle entre les jeux de rôle et les médiums traditionnels de fiction (romans, films...).
- Le jeu de rôle prolonge ainsi une tradition plus ancienne de contes où des amis collaborent pour créer une histoire.

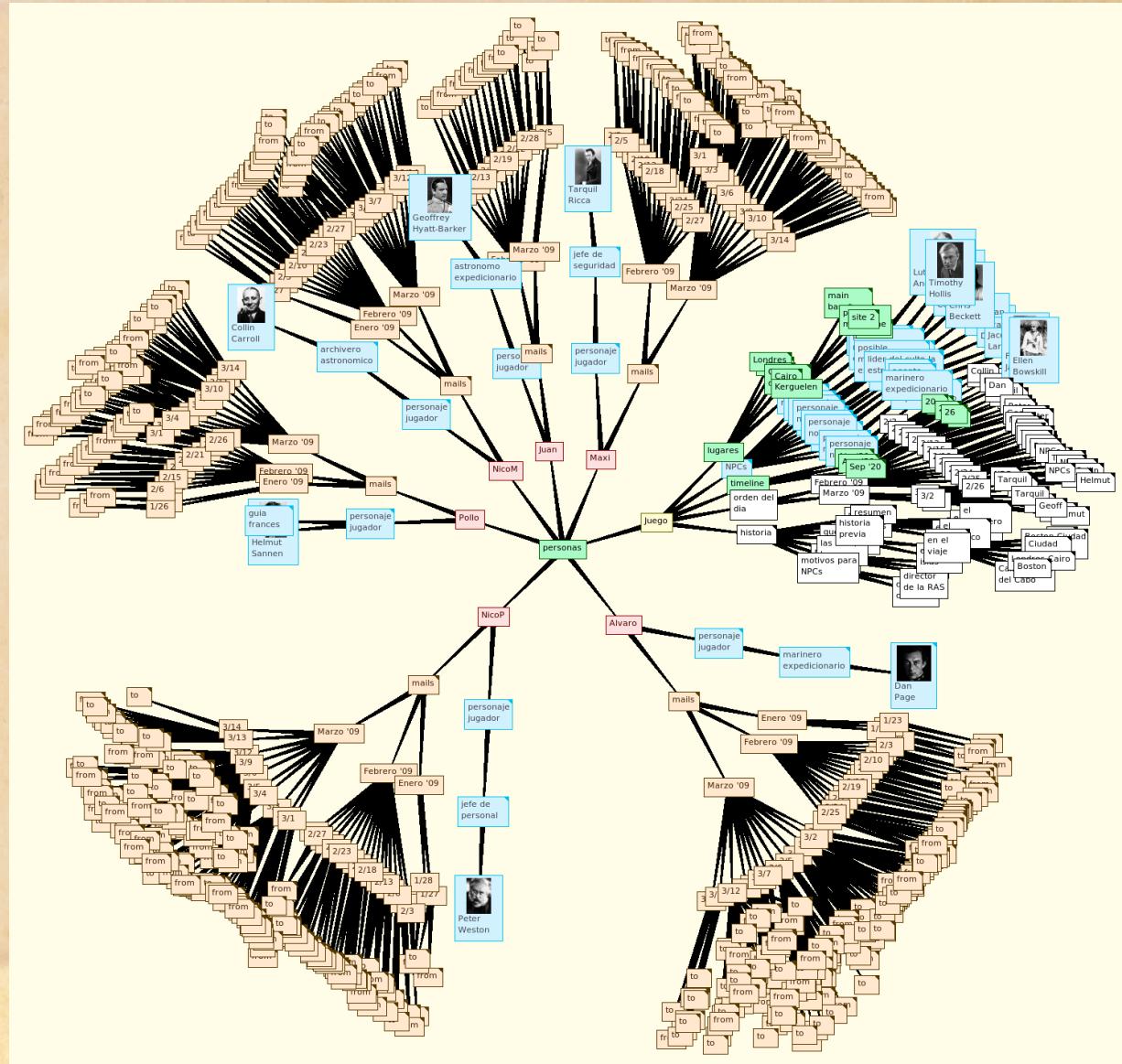
(Source: Wikipedia)

# L'Appel de Cthulhu (jeu de rôle)

- Un jeu de rôle créé aux États-Unis au début des années 1980 par Sandy Petersen.
- Le titre vient de la nouvelle du même nom écrite par l'écrivain fantastique américain **H. P. Lovecraft**.
- Emprunte à l'univers de H. P. Lovecraft son ambiance mystérieuse et oppressante (le Mythe de Cthulhu) qui décrit un ensemble de puissances maléfiques, incarnations de forces cosmiques et primitives : les Grands Anciens.
  - Beaucoup de ces créatures sont aquatiques (et bien sûr, magiques !)
- Les joueurs sont confrontés à des situations où ils doivent déjouer les complots de ces créatures et de leurs adorateurs.
  - Les personnage non-joueur sont tous fous et dangereux !

(Source: Wikipedia)

# Pourquoi Poisson Magique ?



# The Lamson MVC Mail Framework

- A quick overview of Zed Shaw's project
  - <http://lamsonproject.org>
- Main example:
  - <http://librelist.com>
- Philosophy:
  - Modern (python, Unicode, sane)
  - MVC
  - Based on FSM



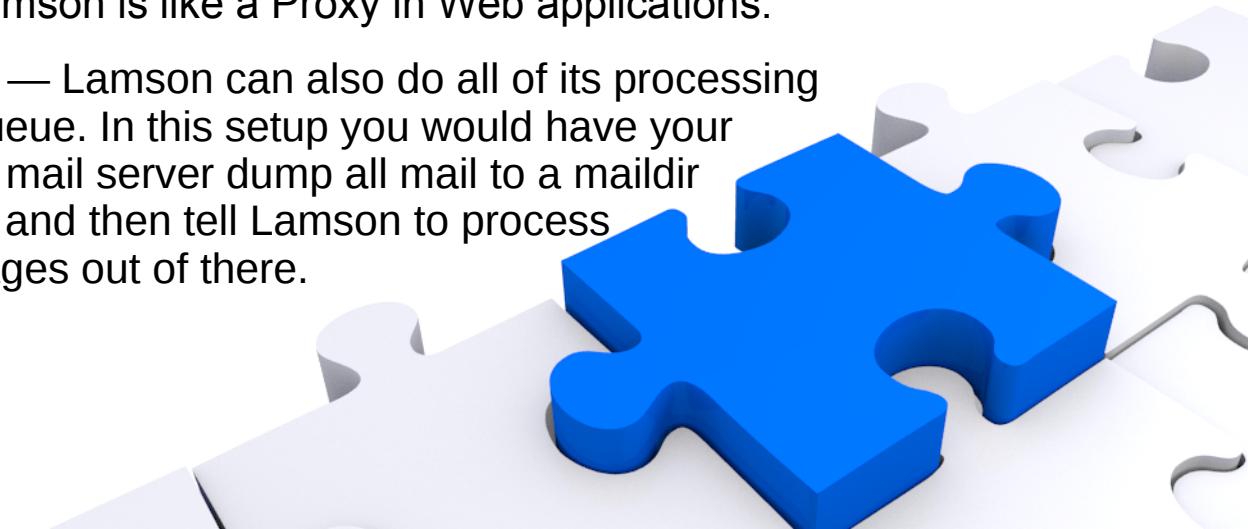
# Lamson 30 sec Introduction

```
$ easy_install lamson
$ lamson gen -project mymailserver
$ cd mymailserver
$ lamson start
$ lamson log
$ nosetests
$ lamson help -for send
$ lamson send -sender me@mydomain.com -to
test@test.com \
    -subject "My test." -body "Hi there." -port 8823
$ less logs/lamson.log
$ mutt -F muttrc
$ lamson stop -ALL run
```



# Important Terminology

- **MVC** — design methodology used in web application frameworks where the data (model), presentation (view), and logic (controller) layers of the application are strictly separated.
- **FSM** — Lamson uses the concept of a Finite State Machine to control how handlers execute. Each time it runs it will perform an action based on what it is send and what it was doing last.
- **Template** — Lamson generates the bodies of its messages using Templates, which are text files that have parts that get replaced with variables you pass in. Templates are converted to their final form with a process called rendering.
- **Relay** — The relay for a Lamson server is where Lamson delivers its messages. Usually the Relay is a smart tougher server that's not as smart, but very good at delivering mail.
- **Receiver** — Lamson typically runs as the Receiver of email.
  - Lamson runs “in front” of an SMTP server like Postfix. It listens on port 25, handles the mail it should, and forwards the rest to the Relay.
  - Lamson is like a Proxy in Web applications.
- **Queue** — Lamson can also do all of its processing off a queue. In this setup you would have your normal mail server dump all mail to a maildir queue, and then tell Lamson to process messages out of there.



# Directory Structure of a Lamson Project

```
lamson/data/prototype
├── app
│   ├── handlers
│   │   └── __init__.py
│   │   └── sample.py
│   └── __init__.py
│       └── model
│           └── __init__.py
├── config
│   ├── boot.py
│   ├── __init__.py
│   ├── logging.conf
│   ├── settings.py
│   ├── testing.py
│   └── test_logging.conf
├── muttrc
└── README
└── tests
    ├── handlers
    │   └── __init__.py
    │   └── open_relay_tests.py
    ├── __init__.py
    └── model
        └── __init__.py
            └── templates
                └── __init__.py
8 directories, 16 files
```

- App folder with
  - Models
  - Handlers
- Config folder
- Test folder

*Django flavor?*



# The Librelist project

- A mailing list without maintainer
- Send an email to [begonias@librelist.com](mailto:begonias@librelist.com)
  - Librelist will reply saying the list doesn't exist and ask for confirmation for creation
  - Upon confirmation, you're automatically subscribed
- Subscribing happens on first email to the list
  - Which is then sent to the list unless is trivial
- Web interface, rsync access, etc



# Directory Structure of Librelist Project

```
examples/librelist
├── app
│   ├── handlers
│   ├── model
│   └── templates
│       └── mail
├── config
├── deploy
│   ├── env
│   ├── lib
│   ├── migrations
│   └── scripts
└── lib
    └── tests
        ├── handlers
        ├── model
        └── templates
└── webapp
    └── librelist
        └── migrations
19 directories
```

- App folder with
  - Templates!
- Deploy folder
- Webapp folder
  - Django App
  - Some minimal settings magic to make it play ball with the folders



# Handlers (FSMs)

- Routes
- States are functions
- Stateless vs. regular “states”

```
@route("roll-(hashid)@(host)", hashid="[0-9]+")
def START(message, hashid=None, host=None):
    hashid = int(hashid)
    roll = find_roll(hashid)
    if roll is None:
        logging.info("Unknown roll %d" % (hashid))
    return
```



# Poisson Magique: VM deployment

- The documentation discusses three different ways to deploy Lamson
- My own setting is different as I'm running P. M. in a VM with port 25 already taken
- The host machine is a Debian GNU/Linux box running exim4
  - /etc/exim4/hubbed\_hosts
    - poissonmagique.net: 10.10.0.55::2525



# Background Queues

- The main thread that listens to port 25
- Background queues instead monitor a folder
  - In Maildir format
  - They wake up every a specified number of seconds and process all new messages in the folder
  - The handlers are the same as in the main thread
  - Although they need to be @stateless



# Unicode Support

- A most advertised feature
- It didn't work the way I expected initially
  - Some methods return bytestrings encoded in UTF-8
  - Strings in python 2 are problematic
- Lamson does a gargantuan job at handling encoding in emails
  - The most dense code in Lamson are the heuristics to guess encodings
  - Use it or you'll have to reinvent the wheel in similar ways



# Key Features I haven't Used

- Double Opt-In
  - The capability to track subscriptions has its own page on the site
  - It is particularly important given the new Canadian regulations regarding spam.
- HTML templates
  - The framework takes care of in-lining the CSS as required by the arcane email HTML rendering engines
  - Zed hates HTML email. Me too.



# P. M. : Conception du logiciel

- Deux processus
  - Processus principal, executé avec l'arrive de chaque courriel
  - Processus de téléchargement dans la base de données
- Revue dans le site Web
  - Écrit avec Django et Bootstrap
- Conception des modèles de données
  - Human (a un courriel extérne)
  - Character (a un courriel intérne)
- Chaque fois un nouveaux courriel est arrivé...
  - On verifié l'expéditeur. Es-que nous lui/le connais?
    - Non? On l'ignoré
  - On verifié si l'expéditeur est un GM
    - Les courriel des GMs sont envoyés tout suite
  - On verifié si le destinataire est un GM
    - Ainsi, envoyé tout suite
  - Sinon, envoyé un notice au GM
    - Tiens courriel à la modération

# P. M. : Jettent des dés

- Secure Dice
  - Site internet indépendant
  - Jettenet de dés à courriel
  - <http://www.rpglibrary.org/software/securedice/>
  - P. M. donner chaque joueur un lien d'internet personnalisé
    - Copie Conforme à un adresse électronique unique  
(dice-<id>@poissonmagique.net)
- Dans l'avenir:
  - Avoir des commandes spéciales sur les courriels pour créer les jets de dés.
    - Seulement pour les courriels écrit pour des directeurs du jeux.

# P. M. : Jettent des dés



.poissonmagique.net    Roll a dice    Messages ▾    Poisson Magique ▾

---

**Repeats:**

**Sides:**

**Add:**

**Set repeat:**

**Description:**

**What:**

**Who:**  ▾

---

Poisson Magique, Copyright (C) 2013 by [Pablo Duboue](#).  
This software comes with NO WARRANTY and it is available under the terms of the [AGPL-3.0 license](#).

# P. M. : Jettent des dés



- <http://poissonmagique.net/dice/roll/895930>
- <http://www.rpglibrary.org/software/securedice/?dq=2&ds=6&dm=2&dt=1&cc=roll-895930%40poissonmagique.net&to=pablo.duboue%40gmail.com&sub=2D6%2B2+%282D6%2B2+fall+injury%29>

## Secure Dice

Brandon Blackmoor

Free (GPL) ([source](#), 2012-11-17)

RPG Library Secure Dice is a free online dice roller which will generate random numbers, generate a MD5 checksum of the results, and email those results to the email address(es) you specify.

34,368,503 dice rolled since 2005-11-06.

Roll  d  +  , and .  
then roll and subtract  d  +  , and .

OR...

Roll  Fudge (or FATE) dice

(If you select Fudge dice, the section above will be ignored.)

Roll this set of dice  times.  Sort dice sets?

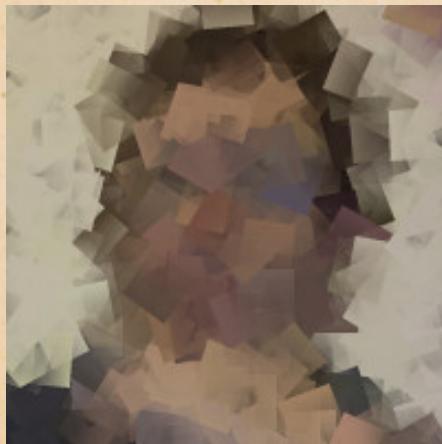
Send the signed results of this roll to yourself:   Also send email **from** this address?

and the GM:

with this subject:

# P. M. : Joue en cours !

- Sous-marine  
20 personnages  
5-15 personnes (!)



# Commentaires finales

- Le cadre de logiciel "Lamson" est très agréable
  - Essayez-le dans votre prochaine projet !
- Poisson magique est un travaux en cours
  - Si vous s'intéresse, collabore à github:
  - <https://github.com/DrDub/poissonmagique>

Connais vous le defis <http://24pullrequests.com> ?



This work is licensed under a Creative Commons  
Attribution-ShareAlike 3.0 Unported License.