

AI match maker for Prologin finals

Thibault Allançon
December 2018

Background

- Non-profit organization
since 1991



- Non-profit organization since 1991
- National computer science contest

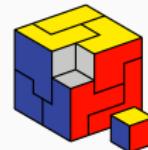


- Non-profit organization since 1991
- National computer science contest
- Girls Can Code!



Prologin

2019



- Free computer science contest
- Open to French-speaking students under 21
- Introduces the world of programming and algorithms

National Computer Science Contest

▪ Qualification

The screenshot shows a web page for the Prologin Qualification 2019 competition. The top navigation bar includes links for "Participer", "Découvrir", "S'entraîner" (which is highlighted in blue), "Archives", "Forums", "Connexion", and "S'inscrire". Below the navigation, a breadcrumb trail shows the path: "Prologin > Entrainement > Qualification2019 > Statuettes". The main content area has a title "Statuettes – Qualification 2019" and a "Niveau 5" badge. To the right, there are four links: "Énoncé" (Statement), "Contraintes d'exécution" (Execution constraints), "Exemples d'entrée/sortie" (Input/output examples), and "Proposez votre solution" (Propose your solution). The "Énoncé" section contains the problem statement: "C'est maintenant la fin des vacances pour Haruhi et Joseph, Haruhi est à l'île de Pâques et cherche un cadeau pour son groupe d'amis. Elle découvre une guirlande composée de statuettes de l'île, de différentes tailles. Son groupe d'amis a une photo emblématique et pour y faire un clin d'œil elle aimerait avoir un bout de cette guirlande où chacune des statuettes représente un des amis du groupe. Les amis sont reconnaissables par leurs tailles relatives respectives." The "Entrée" section describes the input format: "L'entrée comprendra :". A bulleted list follows:

- Deux entiers, n le nombre d'amis sur la photo, et m la taille totale de la guirlande (le nombre de statuettes sur la guirlande).
- Sur la ligne suivante, n entiers $1 \leq p_i \leq n$ donnent la position du i -ème ami le plus petit. Ces tailles forment une permutation de $1, \dots, n$.
- Si $p_i = k$ alors l'ami en position k est le i -ème plus petit des n amis sur la photo.
- Sur la ligne suivante m entiers h_j , la taille de la statuette en position j .

Online exercises and quiz (3 months)

National Computer Science Contest

- Qualification
- Regional events



Theoretical and practical exam (1 day)

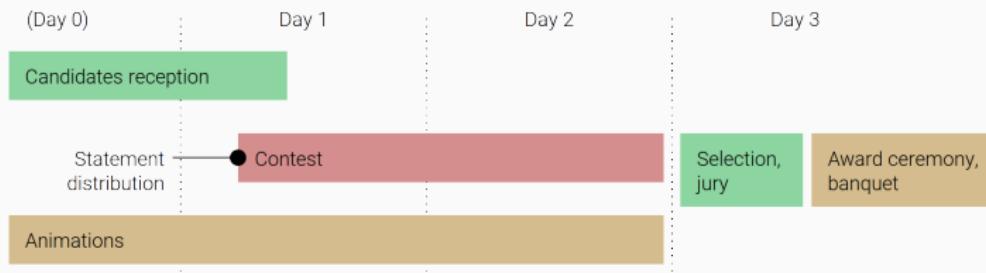
National Computer Science Contest

- Qualification
- Regional events
- Finals



Artificial intelligence battle (3 days)

The Finals



The Game

We create a custom game every year.



Our needs

- Flexible
- Polyglot
- Distributed
- Isolated

Our solution

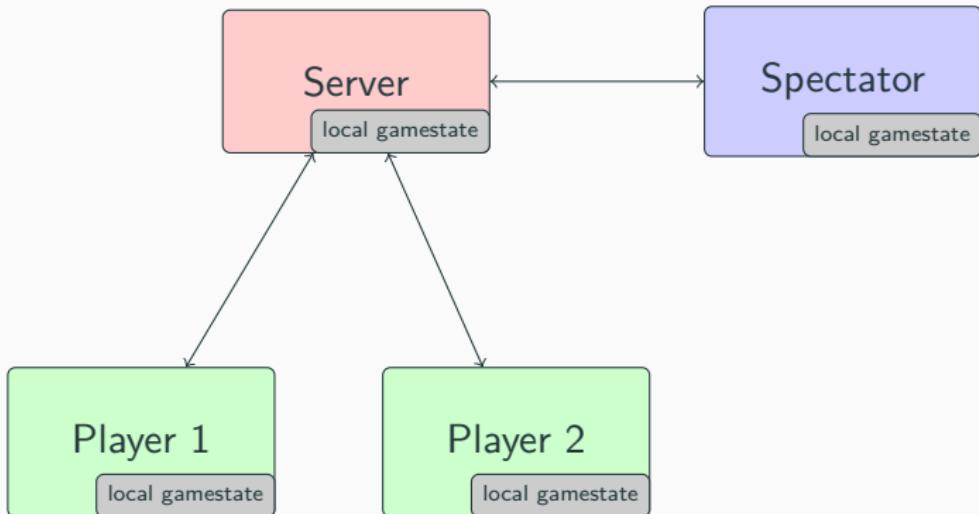
stechec

Our solution

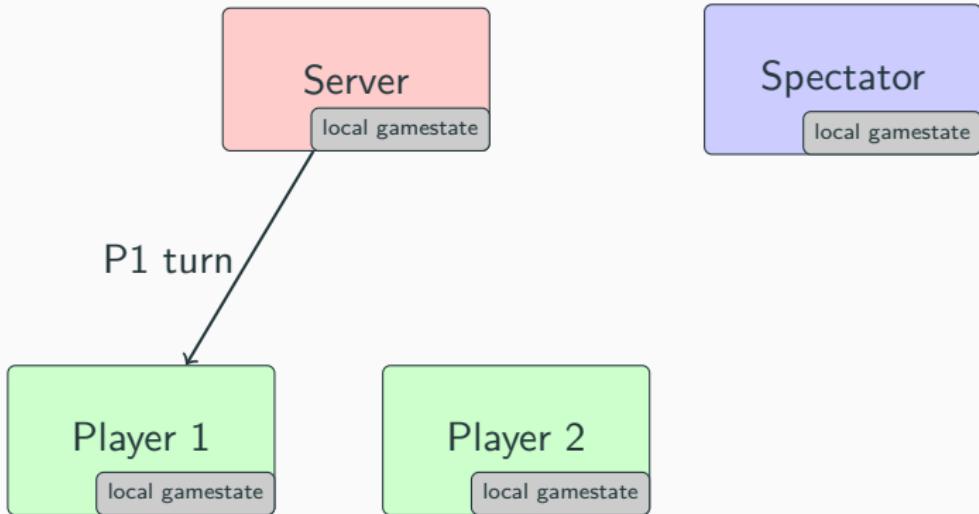
stechec2

Architecture

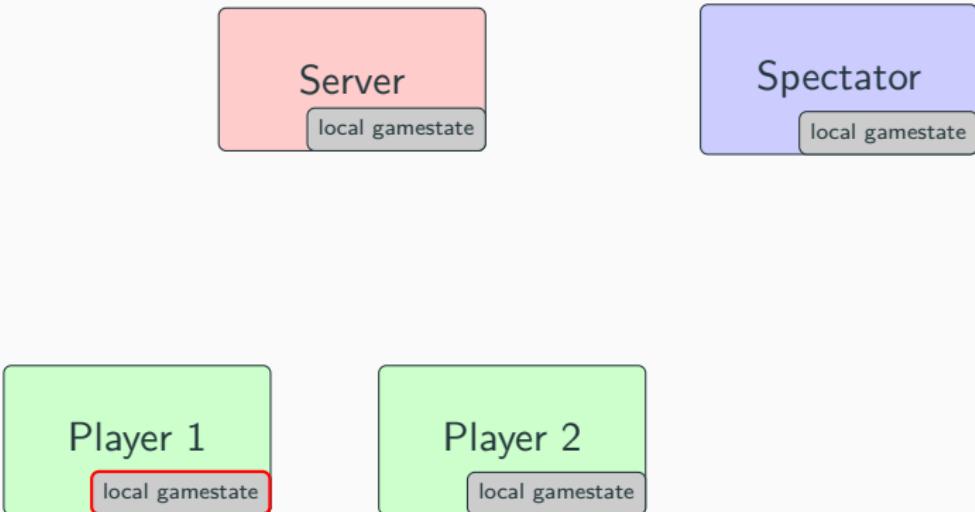
stechec2 architecture



stechec2 architecture

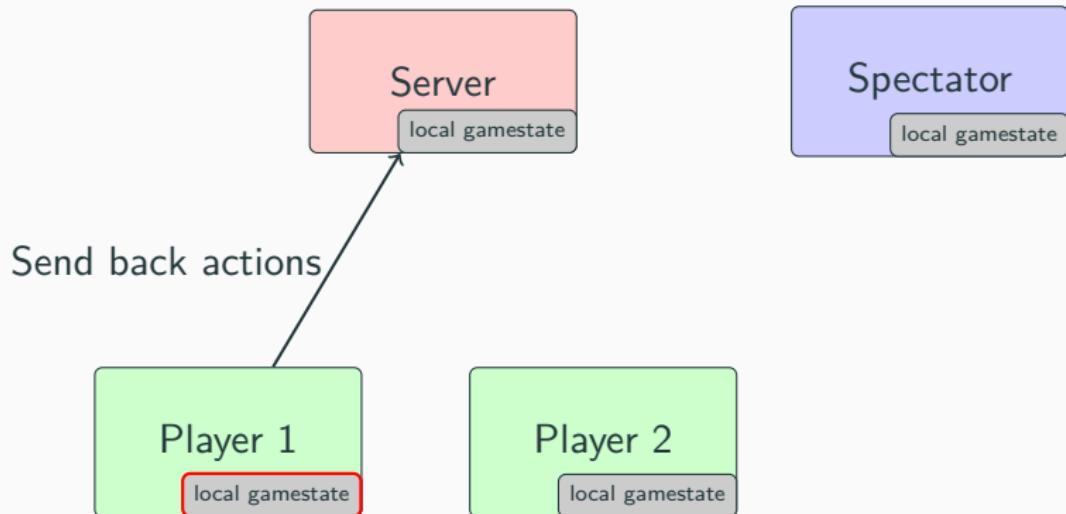


stechec2 architecture



- Call user functions
- Check actions on
local gamestate

stechec2 architecture



stechec2 architecture

Check actions on
local gamestate



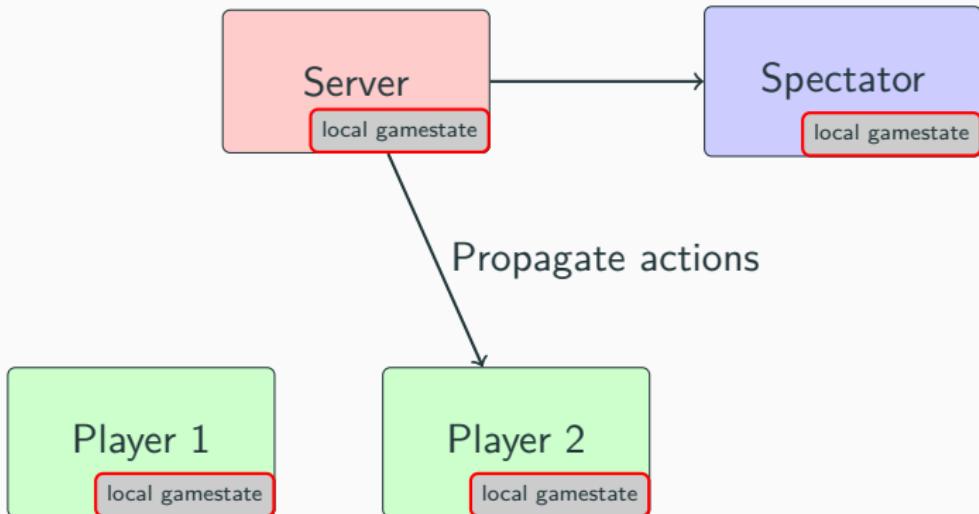
Player 1



Player 2



stechec2 architecture



Problem I: Flexibility

Flexibility

We want generic and re-usable code...

Flexibility

We want generic and re-usable code...

...but we need flexibility to have original games

Interface

Generic interface:

- at_start
- start_of_player_turn
- end_of_round
- ...

Interface

Generic interface:

- `at_start`
- `start_of_player_turn`
- `end_of_round`
- `...`

Each type of rules overloads interface functions:

- Synchronous rules
- Turn based rules

Creating a game

Coding a game always starts with a YAML file:

Creating a game

Coding a game always starts with a YAML file:

- Name
- Rules type

Creating a game

Coding a game always starts with a YAML file:

- Name
- Rules type
- Constants
- Enumerations
- Structures

Creating a game

Coding a game always starts with a YAML file:

- Name
- Rules type
- Constants
- Enumerations
- Structures
- Actions functions
- Observers functions

Creating a game

Coding a game always starts with a YAML file:

- Name
- Rules type
- Constants
- Enumerations
- Structures
- Actions functions
- Observers functions

```
stechec2-generator rules prologin2019.yml prologin2019
```

Problem II: Polyglot

The API user interface

Actions functions:

- `deplacer`
- `glisser`
- `pousser`

Observers functions:

- `position_agent`
- `liste_aliens`
- `info_alien`
- `...`

The API user interface

Actions functions:

- `deplacer`
- `glisser`
- `pousser`

Observers functions:

- `position_agent`
- `liste_aliens`
- `info_alien`
- `...`

Problem: This interface is coded in C++

The API user interface

Actions functions:

- déplacer
- glisser
- pousser

Observers functions:

- position_agent
- liste_aliens
- info_alien
- ...

Problem: This interface is coded in C++

Solution: Use foreign function interface (or simply **FFI**)

Contestant environment

```
stechec2-generator player prologin2019 env
```

Contestant environment

```
stechec2-generator player prologin2019 env
env/
├── c/
├── caml/
├── cs/
├── cxx/
├── haskell/
├── includes/
├── java/
├── php/
├── python/
└── rust/
```

Simple case: C++

```
cxx/
├── Makefile
├── prologin.cc
└── prologin.hh
```

```
#include "prologin.hh"

void partie_init()
{
}

void jouer_tour()
{
}

void partie_fin()
{
}
```

prologin.hh

```
#ifndef PROLOGIN_HH
#define PROLOGIN_HH

...
// constants
#define NB_TOURS 100
...

// structures/enums
typedef enum direction {
    NORD,
    EST,
    SUD,
    OUEST,
} direction;
...

// Actions functions
erreur deplacer(int id_agent, direction dir);
...

// Observers functions
int tour_actuel();
...

#endif
```

Using another language

So let's try with Python.

```
python/
├── api.py
├── interface.cc
├── interface.hh
├── Makefile
└── prologin.py
```

Using another language

So let's try with Python.

```
python/
├── api.py
├── interface.cc
├── interface.hh
├── Makefile
└── prologin.py
```

api.py contains basics
(constants, enums, structs)

Using another language

So let's try with Python.

```
python/
├── api.py
├── interface.cc
├── interface.hh
├── Makefile
└── prologin.py
```

interface.cc enables the
Python code to call C++
functions using FFI

Using another language

Few more examples:

Using another language

Few more examples:

```
caml/
├── api.ml
├── interface.cc
├── interface.hh
├── Makefile
├── prologin.ml
└── prologin.mli
```

Using another language

Few more examples:

```
c/
├── interface.cc
├── interface.hh
├── Makefile
├── prologin.c
└── prologin.h
```

Using another language

Few more examples:

```
rust/
├── api.rs
├── ffi.rs
├── interface.cc
├── interface.hh
├── Makefile
├── prologin.h
└── prologin.rs
```

```
haskell/
├── Api.hs
├── CApi.hsc
├── interface.cc
├── interface_c.cc
├── interface.hh
├── Makefile
└── Prologin.hs
```

Fun fact

Rust and Haskell were added by contestants themselves!

Building

All the languages come with generated code **and** a Makefile

Building

All the languages come with generated code **and** a Makefile

C++

```
$ make
cxx      prologin.cc -> prologin.o
lib      champion.so
```

Building

All the languages come with generated code **and** a Makefile

Python

```
$ make
cxx      interface.cc -> interface.o
lib      champion.so
```

Interpreted languages still need to compile the interface

Building

All the languages come with generated code **and** a Makefile

Rust

```
$ make
rustc  prologin.rs -> prologin.o
cxx     interface.cc -> interface.o
lib     champion.so
```

Building

All the languages come with generated code **and** a Makefile

Rust

```
$ make
rustc  prologin.rs -> prologin.o
cxx     interface.cc -> interface.o
lib     champion.so
```

What is this champion.so file anyway?

.so stands for **shared objects**

This is a dynamically linked library, used in the matchmaking system.

Launching a match

```
$ make  
cxx      prologin.cc -> prologin.o  
lib      champion.so  
$ make tar  
tar champion.tgz
```

Upload this compressed file to the local finals website.

Launching a match

The screenshot shows a web application interface for managing champions. On the left, there's a sidebar with navigation links: Accueil, État des serveurs, Champions, Mes champions, Envoyer un champion (highlighted with a red border), Tous les champions, Matches, Mes matches, Les matches des mes champions, Lancer un match, and Tous les matches. The main content area has a title "Envoyer un champion". It contains three input fields: "Nom" (Name) with an empty text input, "Sources" (Sources) with a "Browse..." button and a message "No file selected.", and "Commentaire" (Commentary) with an empty text input. Below these fields are two buttons: "Envoyer le champion" (Send champion) in a dark box and "Envoyer un champion" (Send champion) in a light box. The background features a large, faint illustration of a hand holding a sword.

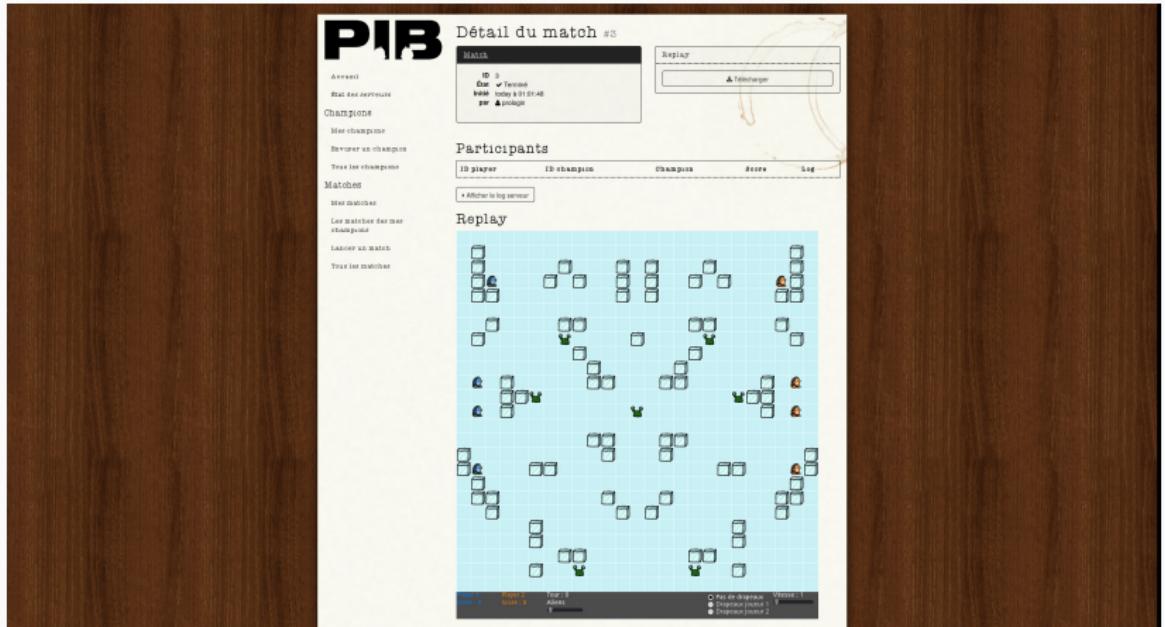
Uploading your champion

Launching a match

The screenshot shows the PIR web application interface. At the top right, there are user navigation links: 'prologin', 'Admin', and '<> API'. The main header 'PIR' is on the left. Below it, the title 'Lancer un match' is centered. Two dropdown menus labeled 'Champion 1' and 'Champion 2' are positioned above a large button labeled 'Lancer le match'. On the left side, there is a sidebar with several menu items: 'Accueil', 'Stat des serveurs', 'Champions' (with sub-options 'Mes champions', 'Envoyer un champion', and 'Tous les champions'), 'Matches' (with sub-options 'Mes matches', 'Les matches des mes champions', and 'Tous les matches'). The 'Lancer un match' button is highlighted with a red border. The background features a faint illustration of a hand holding a small object.

Launching a new match

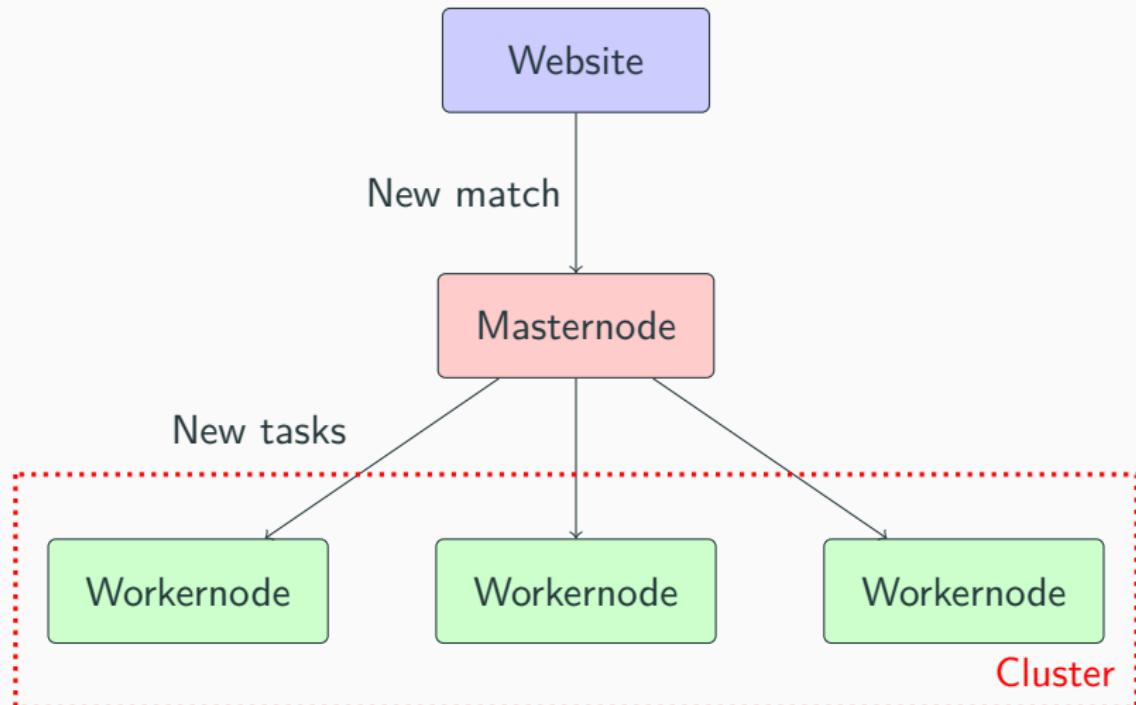
Launching a match



Match replay

Problem III: Distributed

Launching a match II



Tasks

Masternode **divides tasks among workernodes.**

Tasks

Masternode **divides tasks among workernodes.**

A workernode **completes a task.**

(compiling a champion, or running a match)

Network

Communication between masternode and workernodes uses:

- Remote procedure calls (**RPC**)
- Asynchronous methods

Problem IV: Isolated

Need for isolation

- We **do not want** to execute unknown code on our machines
- Candidates have time and memory limits for their AI

Need for isolation

- We **do not want** to execute unknown code on our machines
- Candidates have time and memory limits for their AI

Workernode's tasks are **always** running in isolated mode.

isolate

<https://github.com/ioi/isolate>

*a sandbox built to safely run untrusted executables,
offering them a limited-access environment and
preventing them from affecting the host system.*

isolate

<https://github.com/ioi/isolate>

*a sandbox built to safely run untrusted executables,
offering them a limited-access environment and
preventing them from affecting the host system.*

Built for IOI (International Olympiads in Informatics)



camisole

camisole is:

- Our own secure online judge
- Running isolate as a backend

<https://github.com/prologin/camisole>

Conclusion

Useful infos

<https://prologin.org>

<https://github.com/prologin/stechec2>

<https://github.com/prologin/sadm>

<https://github.com/prologin/prologin2018>

Useful infos

<https://prologin.org>

<https://github.com/prologin/stechec2>

<https://github.com/prologin/sadm>

<https://github.com/prologin/prologin2018>

info@prologin.org

#prologin @ irc.freenode.net

Questions?

Thanks for listening!

Thibault 'haltode' Allançon
thibault.allancon@prologin.org