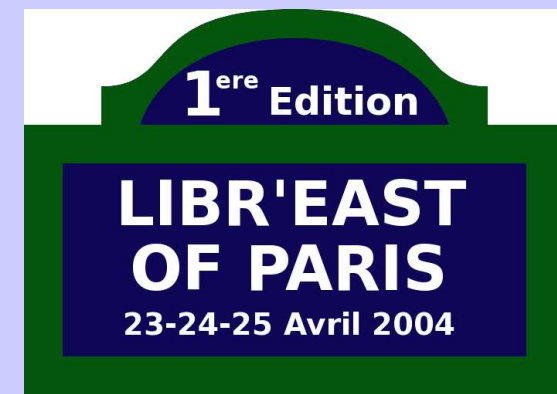


*Le logiciel DemExp  
pour  
le projet d'Expérience Démocratique*

Équipe de développement DemExp  
I. Oger, F. Henry, F. Lehobey, D. Mentré

[savannah.nongnu.org/projects/demexp](http://savannah.nongnu.org/projects/demexp)

[www.demexp.org](http://www.demexp.org)



# *Pourquoi l'Expérience Démocratique ?*

- ◆ La démocratie représentative n'est pas assez représentative

Trop de problèmes à résumer en une seule élection

- ◆ Projet d'Expérience Démocratique

Réfléchir et décider problème par problème

[www.demexp.org](http://www.demexp.org)

# *Projet d'Expérience Démocratique*

- ◆ Projet d'Expérience Démocratique ([www.demexp.org](http://www.demexp.org))
  - x N'importe qui peut poser une question  
Ex: Le logiciel est-il brevetable ?
  - x N'importe qui peut apporter une réponse  
Ex: Oui / Non / Oui pour 2 ans / Oui pour 5 ans / Question rejetée
  - x N'importe qui peut voter
  - x N'importe qui peut déléguer son vote
- ⇒ Extraire à partir des votes la position du groupe  
⇒ un logiciel de vote **spécifique**

# *Fonctionnalités du logiciel DemExp*

◆ Base de Questions  $\Rightarrow$  *base des positions*

◆ Un système de *classification*

Sur chaque question sont attachés des mots-clés (Rennes, écologie, ...)

◆ Un système de *délégation*

Je peux déléguer les questions du mot-clé « logiciel libre » à Bruce Perens, RMS, la FSF Europe, le projet Debian

◆ Une méthode de *vote Condorcet*

Pour choisir une réponse satisfaisante (utilisé par Debian)

# *Base des positions*

## ◆ Une position

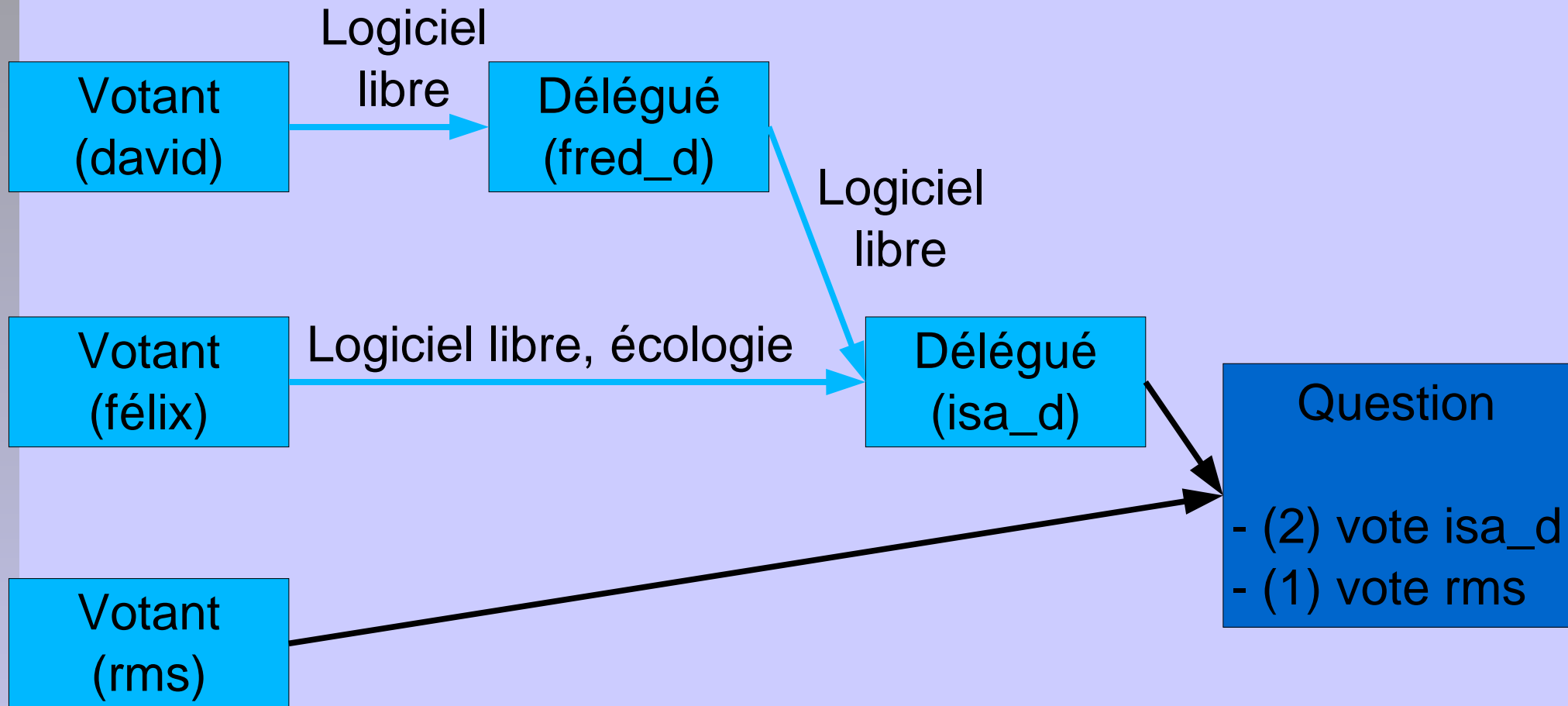
- x une question : [Le logiciel est-il brevetable ?](#)
- x une ou plusieurs réponses (avec un lien : débat)
  1. [Oui](#)
  2. [Oui pour 2 ans](#)
  3. [Non \(http://ffii.org\)](http://ffii.org)
- x des votes
  - [Alan : 3](#) – [Bill : 1, 2](#)

## ◆ Base des positions : regroupement des positions

# Classification

- ◆ Chaque question contient des mots clés  
question : Le logiciel est-il brevetable ?  
mots-clés : Logiciel libre, logiciel, brevet, droit, ...
- ◆ Construire la classification  
⇒ rôle des « responsables classification »
  - x ajouter des mots clés
  - x rendre les questions publiques
  - x gérer une base cohérente de mots clés

# Délégation

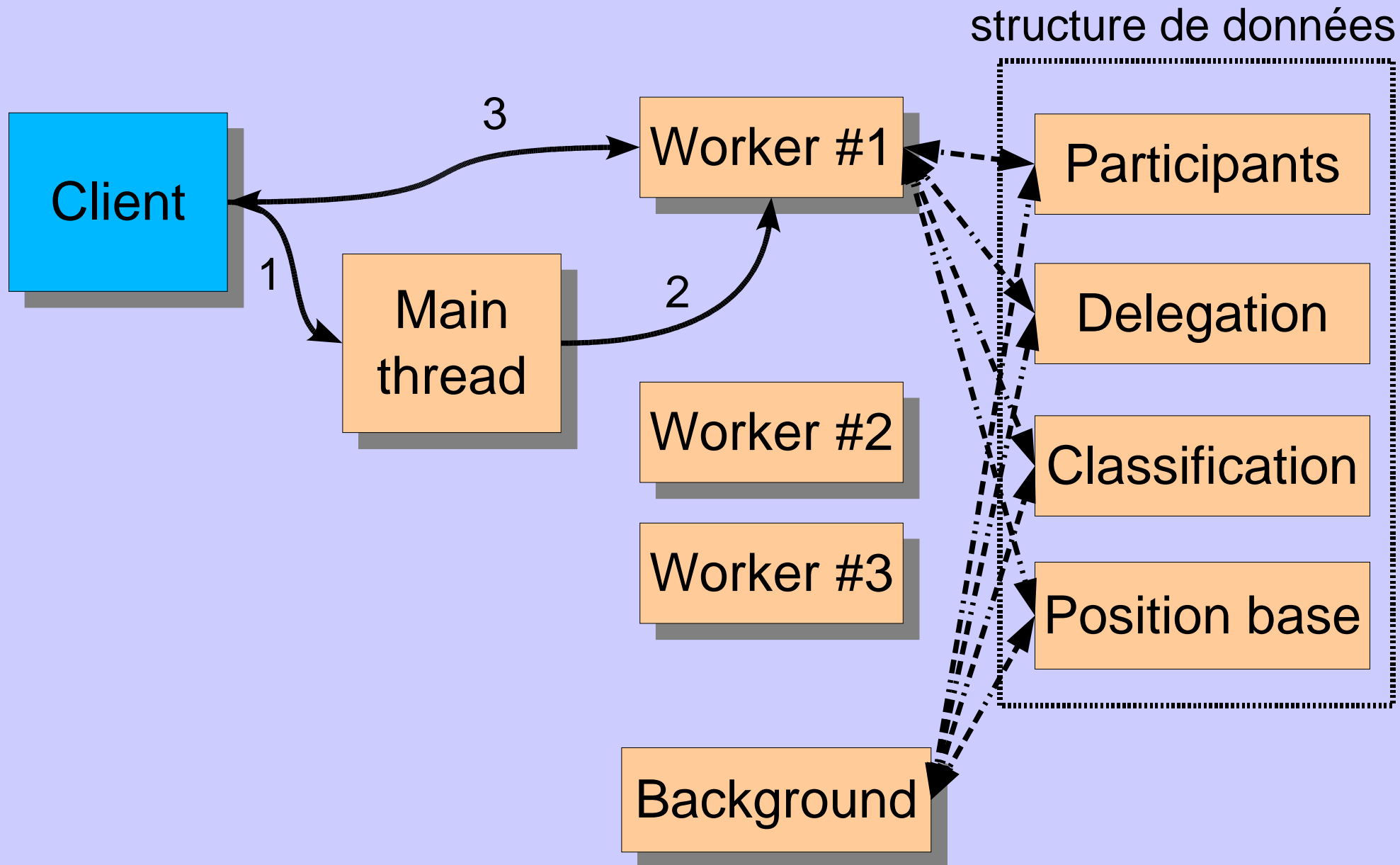


- ◆ Gérer les conflits (hiérarchie personnelle)
- ◆ Éviter les incohérences

# *Architecture de DemExp*

- ◆ Architecture Client/Serveur classique
  - serveur multi-threadé, pas de vraie base de données pour l'instant
- ◆ Un client texte et (une idée de) client graphique
- ◆ Utilisation de standards
  - ✗ XDR (RFC 1832) : encodage réseau, utilisé par NFS/NIS
  - ✗ XML : stockage

# Organisation du serveur



# *Programmation de DemExp*

- ◆ Client et serveur programmés en Objective Caml  
rapidité du C, sûreté de l'Ada, souplesse du Lisp  
« *The programming tool of choice for discriminating hackers* »
- ◆ Avoir les meilleures pratiques de programmation
  - x code documenté : programmation littéraire (noweb),  
comme dans T<sub>E</sub>X
  - x tests - ? preuves formelles ?
  - x gestion de versions : GNU Arch (tla)
  - x c'est un *objectif* : l'état actuel est encore loin d'être idéal
- ◆ Logiciel libre : licence GNU GPL

# Exemple de code littéraire (1/2)

## ◆ Code OCaml du serveur

### 11.3 Response handling

When we add a response `r_desc` from `author` to a question `q_desc`, we first check that this response is not already present. In case the response has already been seen, we raise `Response_already_made`.

The external link (`extlink`) could be provided at a later time.

In case the question `q_desc` cannot be found, we raise `Question_not_found`.

**fixme:** questions should be normalized before being compared.

FIXME

```
71d <posbase.ml 70a>+≡ <71c 72a>
let add_response ~q_desc ~r_desc ~author ?(extlink = None) () =
  try
    let q = Hashtbl.find !the_position_base q_desc in
      (* look for the same response *)
      let len = Array.length q.responses in
        for i = 0 to len - 1 do
          if q.responses.(i).r_desc = r_desc then raise Response_already_made
        done;
      (* we have a new response so add it *)
      let new_resp = {r_desc = r_desc;
                     r_author = author;
                     external_link = extlink;} in
        let copy_resp i = if i < len then q.responses.(i) else new_resp in
          q.responses <- Array.init (len + 1) copy_resp
        with Not_found ->
          raise Question_not_found
```

# Exemple de code littéraire (2/2)

## ◆ Format des messages réseau en XDR

**Get details on a question** With following message `get_question_details_t`, we ask the server to send us all the information related to question `on_question`.

```
20c  <messages.xdr 15>+≡ <20b 20d>
      struct get_question_details_t {
        question_t on_question;
      };
```

**All details of a question** Message `question_details_t` gives back to the client needed information on a question for a vote:

- the `question_author`
- the `question_limit_date`
- the list of `available_responses`
- the optional `winning_response`

```
20d  <messages.xdr 15>+≡ <20c 20e>
      typedef int winning_responses_t<MAX_NUMBER_WINNING_RESPONSES>;

      struct question_details_t {
        login_t question_author;
        date_t *question_limit_date;
        response_t available_responses<MAX_NUMBER_RESPONSES>;
        winning_responses_t *winning_responses;
      };
```

# État de DemExp (v0.2)

## ◆ Serveur minimal

- x *Fait* : questions, délégation, vote Condorcet, classification (hiérarchique), bibliothèque de communication, auto-tests au travers du réseau, lecture/écriture des bases, multi-threading
- x *Manquent* : droits d'accès, sécurité, import/export XML

## ◆ Client texte rudimentaire...

...voire très rudimentaire !

## ◆ Client graphique

que des boîtes de dialogue, pas de réseau derrière

# Objectifs à court terme (v0.3)

- ◆ Re-conception du système de classification
  - utiliser des mots clés plutôt qu'une hiérarchie (v0.2)
  - ⇒ refaire la délégation
- ◆ Gestion des nouvelles questions (file d'attente)
  - des « responsables classification » ajoutent les mots clés
- ◆ Droits d'accès
  - x distinguer administrateurs / classificateurs / utilisateurs
  - x droits d'accès ≠ capacité à modifier bases et votes
- ◆ Un client texte et un client graphique utilisables

# *Futur de DemExp (1/2)*

- ◆ Pour la v0.4 : version fonctionnelle  
pour tester dans de petits groupes (LUGs, associations)
- ◆ Pour la v0.6, concevoir et réaliser :
  - x sécurité et cryptographie
    - probablement un protocole spécifique (ébauche disponible)
    - réutilisation de bibliothèques reconnues ?
  - x internationalisation
    - affichages des questions et réponses dans une langue
    - infrastructure de traduction (existant ?)

# *Futur de DemExp (2/2)*

## ◆ Pour la 1.0 : avoir plusieurs clients **graphiques**

Interface graphique **primordiale** ⇒ l'ergonomie a un impact direct sur la viabilité du projet

## ◆ Et après...

- x *scalability* : 100 milliards de questions, 5 milliards de votants, serveurs dupliqués, ...  
« assez de vote pour chaque atome dans l'univers » :-)
- x classification améliorée (LISFS de l'Irisa)
- x infrastructure d'information des mises à jour

# *Si vous êtes intéressés...*

- ◆ [savannah.nongnu.org/projects/demexp](http://savannah.nongnu.org/projects/demexp)
- ◆ Projet *très* large
  - à terme, *tous* les domaines de l'informatique :  
classification, systèmes distribués, bases de données,  
interfaces utilisateur, sécurité, réseau, ...
- ◆ Nous sommes intéressés par toutes les solutions  
langages (OCaml pas obligatoire), interfaces utilisateur,  
conceptions
- ◆ Apportez nous votre expérience et expertise  
aidez nous à faire un système parfait

# *En guise de conclusion*

- ◆ Il y a du code qui marche  
mais on va en casser une partie
- ◆ Un projet ambitieux  
des défis techniques pour des défis politiques
- ◆ On recherche du monde
- ◆ Vive la démocratie :-)

[savannah.nongnu.org/projects/demexp](http://savannah.nongnu.org/projects/demexp)

[www.demexp.org](http://www.demexp.org)

# *Questions ?*



**(on vote ?)**