

UDebate

Principe

Pouvoir réellement débattre **efficacement** avec un nombre considérable de personnes grâce à un système de vote et de mise en valeur intelligent et décentralisé.

Décentralisation

Pour lutter contre les éventuels dérives de censure que l'on connaît aujourd'hui, il est important que l'architecture du système soit décentralisée, que chacun possède son site de débat.

- N'importe qui peut héberger un serveur **UDebate** avec ses débats et les commentaires qui vont avec.
- On peut renseigner au serveur d'autres instances **UDebate**, ce qui lui permet de récupérer les sujets les plus populaires et leurs commentaires afin de les proposer à ses internautes. Si une personne poste un commentaire, il est immédiatement rappatrié sur le serveur d'origine. *Mesures techniques pour empêcher l'abus à prévoir.*

Interface

Une interface digne de ce nom est à réfléchir. C'est le cœur même du site, une simplicité d'utilisation sans pareil est nécessaire.

- Simplicité d'utilisation
- Afficher les arguments de manière intelligente avec des systèmes de vote
- Les plus gros arguments doivent facilement être lisibles par l'internaute du premier coup (*pas comme Reddit*) (Proposition : d'un côté les arguments pour et de l'autre les contre. Mais pas terrible : tout n'est pas si binaire)
- Anonymat indispensable, un espace membre ne serait sans doute pas nécessaire

Solutions techniques

Décentralisation

- Chaque serveur a ses débats et ses commentaires
- Il a aussi une liste d'autres instances de **UDebate** pour proposer plus de débats à ses visiteurs
- Une fois toutes les **n** minutes les débats sont récupérés sur les autres serveurs

- Au chargement d'un débat provenant du **serveur 2** sur un **serveur 1** par un client, les commentaires sont chargés sur au client par le serveur via l'API du **serveur 2**.
- Le **serveur 1** est connecté par socket au **serveur 2**
- Lorsqu'un commentaire est laissé sur le **serveur 1**, il envoi en temps réel via le socket le nouveau commentaire sur le **serveur 2** qui va l'enregistrer définitivement et va transmettre, via socket toujours, celui-ci aux serveurs connectés à lui.

Question : est-ce que les commentaires sont enregistrés sur le **serveur 1** ou sont juste récupérés à titre temporaire comme ci-dessus ?

Ça peut-être très intéressant car comme tous les serveurs sont interconnectés via socket, ils sont tous des miroirs en temps réel. Si un serveur vient de se créer, il récupère tous les débats et les commentaires via l'API puis se connecte en temps réel.

Licence

Aurais-je besoin de préciser que tout le code est ouvert et libre sous licence GPL ?