WotWizard : Mode d'emploi

WotWizard construit, à partir de la chaîne de blocs et de la piscine de Duniter, une prédiction des futures entrées des candidats dans la toile de confiance (TdC, ou Wot en anglais) de Duniter. Il utilise pour cela une simulation du mécanisme de Duniter. Lorsque plusieurs dates d'entrées sont possibles pour la même personne, chacune est affectée d'une probabilité. La liste affichée est mise à jour automatiquement toutes les cinq minutes, et tout changement est signalé par une marque visible.
WotWizard contient aussi plusieurs outils complémentaires et un **Explorateur de Toile de Confiance** rapide.

Attention : Il est nécessaire d'utiliser la version 1.7.17 de Duniter ou une version ultérieure de la série 1.7.x.

Comment l'utiliser ?

Ce programme doit être utilisé sur un ordinateur faisant tourner un nœud **Duniter**.

Le nœud **Duniter** doit être configuré pour WotWizard. Pour cela, éditez son fichier de configuration ("conf.json", dans son répertoire ; voir *Configuration* ci-dessous) en mettant la valeur du champ "storage.wotwizard" à "true".

WotWizard est formé d'une partie serveur (**wwServer**) écrite en *Go* (v1.12.5) et communiquant par fichiers contenant des requêtes *GraphQL* en entrée et des réponses JSON en sortie, et d'une partie "interface utilisateur" facultative (**WotWizard.exe**) écrite en *Component Pascal* (v1.7.1 sous Windows ou Wine), qui crée les requêtes GraphQL par menus et fenêtres interactives et qui affiche les

réponses sous forme de textes ou de graphiques.

wwServer, uniquement sous Linux pour l'instant, peut être lancé dans un répertoire initialement vide. Il crée lui-même les sous-répertoires et fichiers dont il a besoin, le tout restant contenu dans le répertoire initial. La position de la base de données de Duniter, si elle n'est pas standard, peut lui être indiquée au premier lancement par une option de la ligne de commande.

WotWizard.exe tourne nativement sous Windows. Pour fonctionner avec wwServer, il doit être placé dans le même répertoire. Si votre ordinateur tourne sous Linux (nécessaire pour l'instant), installez d'abord "wine".

Mode Serveur

Vous pouvez utiliser WotWizard uniquement comme serveur, si vous voulez assurer vous-même l'affichage de ses résultats, par exemple dans une page web. Après avoir configuré Duniter pour WotWizard (voir ci-dessus), placez l'exécutable **wwServer** dans un répertoire vide. Dans le cas où l'installation de Duniter est standard (sa base de données SQLite doit se trouver dans le fichier "~/.config/duniter/duniter_default/wotwizard-export.db"), il suffit de lancer wwServer sans options:

```
$wwServer
```

Sinon, lors du premier lancement, ajouter l'option -du :

```
$wwServer -du <path to Duniter database>
```

Pour les lancements ultérieurs, l'option -du n'est plus utile.

Après une assez courte phase d"initalisation, que vous pouvez suivre sur le fichier "rsrc/duniter/log.txt", les fichiers de requête peuvent être déposés dans le sous-répertoires "rsrc/duniter/Work/Query". Leurs noms n'a pas d'importance et vous pouvez en déposer autant que vous voulez, ils sont immédiatement lus et détruits. Après traitements, les fichiers réponses apparaissent dans le répertoire "rsrc/duniter/Work/Json" ; voir la règle de correspondance entre les requêtes et les noms des fichiers réponses ci-dessous. En cas d'erreur dans un fichier de requête, des messages explicatifs apparaissent dans log.txt et dans Stderr.

Pour arrêter wwServer, créez un fichier vide "Stop.txt" dans le sous-répertoire "rsrc/duniter/Work". **Attention** : un arrêt obtenu par une autre méthode peut endommager la base de données de WotWizard. Cette base est contenue dans le

fichier "rsrc/duniter/System/DBase.data" et une copie de sauvegarde est créée à chaque lancement ; si vous n'êtes pas sûr de son intégrité, vous pouvez la remplacer, une fois le logiciel à l'arrêt, en renommant la dernière copie de sauvegarde "rsrc/duniter/System/DBase.data.bak".

Liste et fonctionnement des requêtes GraphQL

{WWServerStart}: Install a subscription to the update of the WotWizard window. Do nothing if the subscription is already installed (but may change the name of the output json file, see below).

{WWServerStop}: Erase the subscription to the update of the WotWizard window. Do nothing if the subscription is not already installed.

{WotWizardListFile}: Display the WotWizard file.

{WotWizardListPerm}: Display the list of WotWizard permutations.

{IdSearchFind(Hint:"<hint>"){<set of (OldMembers, Members, FutureMembers)>}}: Display the list of identities whose pseudos or public keys begin with <hint> and whose status is old member, active member or newcomer, according to the given set; the list includes pseudos, hashes and status.

{IdSearchFix(Hash:"<hash>"){<set of (Distance, Quality, Centrality)>}}: Display informations for the identity corresponding to <hash> and including, or not, her distance to the wot, her quality and her degree of centrality according to the given set.

{History(Uid:"<uid>")}: Display the history of a member or an old member.

{Parameters}: Display the block 0 parameters of the money.

{IdentitiesRevoked}: Display the list of revoked identities.

{IdentitiesMissing}: Display the list of excluded, but not yet revoked, identities.

{IdentitiesMembers}: Display the list of active identities.

{CertificationsFrom}: Display the list of active certifications, grouped by senders.

{**CertificationsTo}**: Display the list of active certifications, grouped by receivers.

{Sentries}: Display the list of sentries.

{Sandbox}: Display the content of the sandbox, with different sortings.

{QualitiesDist}: Display the distances to the wot of all active members.

{QualitiesQual}: Display the qualities of all active members.

{CentralitiesAll}: Display the degrees of centrality of all active members.

{MemEnds}: Display the ends of validity of memberships for all active members, sorted by dates.

{MissEnds}: Display the dates of revocation of all excluded, but not yet revoked, identities, sorted by dates.

{CertEnds}: Display the dates of loss of the fifth received certification for all active or excluded, but not revoked, identities.

{MembersCountAll}: Display the number of active members, sorted by dates of events (in or out the wot).

{MembersCountFluxAll(timeUnit:<time unit (s)>)}: Display the flux of active members by <time unit>.

{MembersCountFluxPMAll(timeUnit:<time unit (s)>)}:Display the flux of active members by <time unit> and by member.

{MembersFirstEntryAll}: Display the number of first entries into the wot, sorted by dates of events (entries).

{MembersFEFluxAll(timeUnit:<time unit (s)>)}: Display the flux of first entries by <time unit>.

{MembersFEFluxPMAll(timeUnit:<time unit (s)>)}: Display the flux of first entries by <time unit> and by member.

{MembersLossAll}: Display the number of members exiting the wot, minus the number of reentries (losses), sorted by dates of events (in or out the wot).

{MembersLossFluxAll(timeUnit:<time unit (s)>)}: Display the flux of losses by

<time unit>.

{MembersLossFluxPMAll(timeUnit:<time unit (s)>)}: Display the flux of losses by <time unit> and by member.

If an alias is added to a command (e.g. "wwResult" in {wwResult:WWServerStart}), it gives the name of the output json file (here wwResult.json); otherwise, this name is the name of the command itself (e.g. output of {WWServerStart} is in the file WWServerStart.json).

Mode interactif

Ajoutez le fichier "WotWizard.exe" dans le même répertoire que "wwServer".

Lancez WotWizard en double-cliquant sur le fichier "WotWizard.exe", ou avec la commande en ligne, depuis le répertoire choisi (sous Linux) :

```
$wine WotWizard.exe
```

WotWizard.exe ne lance pas wwServer, qu'il faudra lancer indépendamment, avant ou après. Il ne l'arrête pas non plus à sa fermeture. **Attention (rappel)** : n'arrêtez wwServer qu'à l'aide du dépôt d'un fichier "Stop.txt" dans le sous-répertoire "rsrc/duniter/Work" ; voir ci-dessus. Toutefois, la commande "Arrêter le serveur" du menu "Fichier" crée ce fichier "Stop.txt" et arrête wwServer en toute sécurité.

Après l'ouverture de la fenêtre, choisissez votre langue (et, éventuellement vos polices de caractères) dans "Édition -> Préférences...".

Vous pouvez alors lancer des commandes avec les menus de l'interface et voir les résultats dans les fenêtres qui s'ouvrent.

Configuration

Vous devez donner à wwServer deux informations :

- le chemin vers la base de données SQLite de Duniter. En général :
  + sous linux :
    ~/.config/duniter/duniter_default/wotwizard-export.db
  + sous Windows :
    C:\<Utilisateur>\<nom_utilisateur>\.config\duniter\duniter_default\
    wotwizard-export.db

Vous pouvez changer cette donnée en créant (ou en modifiant) le fichier

"rsrc/duniter/init.txt".
- la taille maximale de mémoire en octets (approximativement) qu'il a le droit d'allouer.

Par défaut : 430000000

Vous pouvez changer cette donnée en créant (ou en modifiant) le fichier "rsrc/duniter/parameters.txt". Plus elle est grande, plus les prévisions de wwServer seront précises, mais si elle est trop grande, wwServer peut se bloquer.

La fenêtre WotWizard

Ouvrez la fenêtre WotWizard avec "WotWizard -> Nouvelle fenêtre WotWizard".

Vous pouvez choisir la façon dont la liste est affichée (par noms ou par dates). Sa mise à jour est automatique à chaque apparition d'un nouveau bloc.
Il y a aussi des métadonnées disponibles, analoques au contenu du **Fichier** (voir ci-sessous).

Lorsque la liste change, deux astérisques encadrent le titre, et un bouton "Vérifier" apparaît. En cliquant sur le bouton, on fait disparaître les astérisques et on peut comparer les anciennes et nouvelles listes (par dates et métadonnées) en cliquant sur le bouton "Comparer", ou en utilisant la commande de menu "Édition -> Comparer les textes" (raccourci clavier : F9).

Vous pouvez changer la mémoire maximale allouée par la fenêtre WotWizard. Plus elle est grande, plus les prévisions de WotWizard seront précises, mais si elle est trop grande, WotWizard peut se bloquer. Par défaut, elle vaut 430000000 octets. Pour la changer, utilisez la commande de menu "Édition -> Changer les paramètres".

Fichier (*WotWizardListFile*): WotWizard calcule ses prévisions à partir d'un état des lieux, appelé *Fichier*. Il s'agit d'une liste de *dossiers* d'entrées et de *certifications internes,* triée par dates de disponibilité. Chaque dossier contient le pseudo du nouveau venu et la liste des certifications (externes) qu'il a reçues. Les certifications internes sont émises vers des personnes déjà membres. L'affichage, obtenu par la commande de menu "WotWizard -> Fichier", donne les informations suivantes :

- *Certification interne* : une ligne contenant :
  + le récepteur et l'émetteur de la certification (pseudos) séparés par une flèche (←) ;
  + la date et l'heure de disponibilité de la certification ;
  + entre parenthèses, la date et l'heure limite de validité de la certification ;
  + un résumé de l'état de la certification : OK si elle est disponible et valide, ou KO sinon.

- *Dossier* :
    + une première ligne décrivant le nouveau venu :
        * le nombre de certifications principales (celles qui déterminent la date de disponibilité du dossier, voir ci-dessous) ;
        * son pseudo ;
        * sa date de disponibilité ;
        * entre parenthèses, la date limite de validité de sa demande d'adhésion ;
        * le résultat du calcul de la règle de distance, en pourcentage de membres référents accessibles ;
        * un résumé de l'état du dossier : OK s'il est valide et si les règles de nombres de certifications valides et de distance sont vérifiées, ou KO sinon (la disponibilité des certifications n'est pas prise en compte) ;
    + une ligne supplémentaire par certification externe, identique à celle d'une certification interne, sauf que le récepteur, qui est le nouveau venu, n'est pas mentionné ; ces certifications sont triées par date de disponibilité.

Certifications principales : Pour qu'un dossier soit valide, il faut qu'il contienne un nombre minimal de certifications (*sigQty* = 5), mais aussi que ces certifications lui permettent de vérifier la règle de distance. Du fait de cette règle de distance, il faut parfois plus que *sigQty* certifications. Les certifications principales d'un dossier sont celles qui permettent, au minimum, le respect des deux règles. Par extension, si l'une des deux règles n'est pas respectée, on considère toutes les règles du dossier comme principales. Comme les certifications passent dans l'ordre de leurs dates de disponibilité, on considère, dans cet ordre, les *sigQty* premières règles (ou moins si le dossier est incomplet) ; si elles permettent de respecter la règle de distance, on s'arrête là, sinon on continue avec les certifications suivantes jusqu'à ce que la règle de distance soit satisfaite ou qu'on ait épuisé toutes les certifications du dossier : on a alors trouvé les certifications principales. La date de disponibilité de la dernière certification principale est celle du dossier (il ne sera vraiment disponible à cette date que s'il est valide).

Permutations (*WotWizardListPerm*): La commande "WotWizard -> Permutations" affiche toutes les permutations d'ordre d'entrées prévues par WotWizard avec leurs probabilités. **Attention** : leur nombre peut être très grand et l'affichage très long !

Explorateur de toile de confiance

IdSearchFind & *IdSearchFix*

Ouvrez l'explorateur avec la commande de menu "Toile de confiance -> chercher une identité".

Voir la carte de l'explorateur.

On peut chercher n'importe quelle identité dans la chaîne de blocs ou dans la piscine en tapant ses premiers caractères ou les premiers caractères de sa clé publique dans le premier champ en haut de la fenêtre et en cliquant sur le bouton "Chercher". Les identités correspondantes s'affichent dans la liste en bas de la fenêtre : choisissez celle qui vous intéresse. Des informations apparaissent dans le cadre "Identité", ainsi que les certifications reçues et envoyées dans le cadre "Certifications". Il est possible d'afficher l'identité d'un émetteur ou d'un récepteur de certification en cliquant sur bouton "Voir" correspondant ; on peut revenir en arrière avec les boutons fléchés. On peut aussi afficher un texte d'informations sur les certificateurs et certifiés en cliquant sur l'un des boutons "A" (liste des certificateurs ou certifiés actuels, classés par ordres alphabétiques, par date limite des demandes d'adhésion (ou date limite avant révocation si la personne concernée n'est plus membre - "×" devant le nom) et par dates limites de validité des certifications, et liste de tous les certificateurs ou certifiés ayant existé et non-révoqués et pouvant donc à nouveau soit certifier, soit être certifiés).
Les dates affichées sont la date d'enregistrement du membre dans la chaîne de blocs, les dates limites d'expiration, pour l'inscription du membre et pour ses certifications, et la date de disponibilité de la prochaine certification émise dans le champ "Disponibilité" ; si cette dernière est déjà disponible, elle est précédée d'un point d'exclamation (!). Si le membre a demandé l'arrêt de son adhésion, la date limite de son inscription est précédée d'une croix (×). Dans ce cas, il ne peut plus recevoir de certifications, et, sauf renouvellement de son adhésion, il sera exclu à cette date.

Le degré de centralité *c* d'un membre est le nombre de chemins orientés (certificateur -> certifié) les plus courts auxquels il appartient est dont la longueur est, au plus, *stepMax* (5) plus un depuis le certificateur. Le niveau de centralité *c′* d'un membre est calculé à partir de son degré de centralité *c* par l'expression :

$$c' = 100 \frac{\ln(1+c)}{\max[\ln(1+c)]}$$

où $\max[\ln(1+c)]$ est le maximum de $\ln(1+c)$ sur l'ensemble des membres.

On peut considérer le *niveau de centralité* d'un membre comme son niveau d'implication dans le respect de la règle de distance des membres qu'il a certifiés, sans préjuger de sa capacité à le faire. Par contre, la *qualité* d'un membre est sa capacité à permettre le respect de la règle de distance par ceux qu'il certifie, sans préjuger de son niveau d'implication.
Un membre qui a une qualité supérieure ou égale à *xpercent* (= 80%), assure à lui tout seul le respect de la règle de distance par ceux qu'il certifie.

Outils

Paramètres (*Parameters*): Affiche les paramètres de base de la monnaie.

Identités révoquées (*IdentitiesRevoked*): Affiche les identités révoquées, avec leurs pseudos et clés publiques, et les dates de leurs adhésions.

Adhésions non-renouvelées (*IdentitiesMissing*): Affiche les identités dont l'adhésion n'a pas été renouvelée, avec leurs pseudos et clés publiques, les dates de leurs adhésions et la date limite avant révocation.

Identités (*IdentitiesMembers*): Affiche les identités de tous les membres, avec leurs pseudos et clés publiques, les dates de leurs adhésions et leurs limites de validité.

Certifications depuis... (*CertificationsFrom*): Affiche toutes les certifications présentes dans la chaîne de blocs, triées par émetteurs, avec leurs dates d'inscription.

Certifications vers... (*CertificationsTo*): Affiche toutes les certifications présentes dans la chaîne de blocs, triées par récepteurs, avec leurs dates d'inscription.

Membres référents (*Sentries*): Affiche les identités des membres référents.

Piscine (*Sandbox*): Affiche les identités et les certifications présentes dans la piscine :

1) Identités triées par leurs hashs, avec hash, clé publique, id et date d'expiration
2) Identités triées par leurs clés publiques, avec clé publique et hash
3) Identités triées par leurs pseudos, avec pseudo et hash
4) Certifications triées par les clés publiques de leurs émetteurs, avec clé publique de l'émetteur, hash du récepteur et date d'expiration
5) Certifications triées par les clés publiques de leurs récepteurs, avec hash du récepteur, clé publique de l'émetteur et date d'expiration

Distances (*QualitiesDist*): Affiche les distances de tous les membres à la toile de confiance (proportions des membres référents atteignables en *stepMax* (5) pas au plus), triées par proportions (avec tracé), puis par pseudo.

Qualités (*QualitiesQual*): Affiche les qualités de tous les membres, triées par qualités (avec tracé), puis par pseudo.

Centralités (*CentralitiesAll*): Affiche les niveaux de centralité de tous les membres, triés par niveaux de centralité (avec tracé), puis par pseudo.

Limites des adhésions (*MemEnds*): Affiche les dates limites de validité des adhésions de tous les membres, triées par date.

Limites des adhésions non-renouvelées (*MissEnds*): Affiche les dates limites avant révocation de toutes les identités dont l'adhésion n'a pas été renouvelée, triées par date.

Limites des certifications (*CertEnds*): Affiche les dates limites de validité des adhésions pour manque de certifications de tous les membres, triées par date.

Évolution

Nombre de membres (*MembersCountAll*): Affiche la liste des dates des blocs où le nombre de membres a changé et les nombres n de membres correspondants depuis le début de la monnaie (avec tracé).

Flux de membres (*MembersCountFluxAll*): Variation du nombre de membres par unité de temps (ici: le mois). C'est donc la dérivée du nombre de membres par rapport au temps t : $\frac{\mathrm{d}n}{\mathrm{d}t}$ (avec tracé).

Flux de membres par membre (*MembersCountFluxPMAll*): Flux de membres divisé par le nombre n de membres : $\frac{\mathrm{d}n}{n\,\mathrm{d}t}$, en pourcentage par unité de temps. C'est auss la dérivée logarithmique de n par rapport au temps $\left(\frac{\mathrm{d}\ln(n)}{\mathrm{d}t}\right)$. Lorsque cette valeur reste constante dans le temps, n croît exponentiellement (avec tracé).

Nombre de premières entrées (*MembersFirstEntryAll*): Affiche la liste des dates des blocs où le nombre de membres entrant pour la première fois dans la toile de confiance a changé et les nombres e de premières entrées correspondantes depuis le début de la monnaie (avec tracé).

Flux de premières entrées (*MembersFEFluxAll*): Variation du nombre de premières entrées par unité de temps (ici: le mois). C'est donc la dérivée du nombre de premières entrées par rapport au temps t : $\frac{\mathrm{d}e}{\mathrm{d}t}$ (avec tracé).

Flux de premières entrées par membre (*MembersFEFluxPMAll*): Flux de premières entrées divisé par le nombre n de membres : $\frac{\mathrm{d}e}{n\,\mathrm{d}t}$, en pourcentage par unité de temps (avec tracé).

Pertes (*MembersLossAll*): Affiche la liste des dates des blocs où le nombre de

membres a changé et les différences $l = e - n$ correspondantes entre les premières entrées e et les nombres de membres n depuis le début de la monnaie (avec tracé).

Flux de pertes (*MembersLossFluxAll*): Variation des pertes par unité de temps (ici: le mois). C'est donc la dérivée des pertes par rapport au temps t : $\frac{\mathrm{d}\,l}{\mathrm{d}\,t}$ (avec tracé).

Flux de pertes par membre (*MembersLossFluxPMAll*): Flux de pertes divisé par le nombre n de membres : $\frac{\mathrm{d}\,l}{n\,\mathrm{d}\,t}$, en pourcentage par unité de temps (avec tracé).

Use it and enjoy! - ¡Úsalos y disfrútalos! - Bonne utilisation - Приятного использованиян - Powodzenia - Viel Spaß

Gérard Meunier