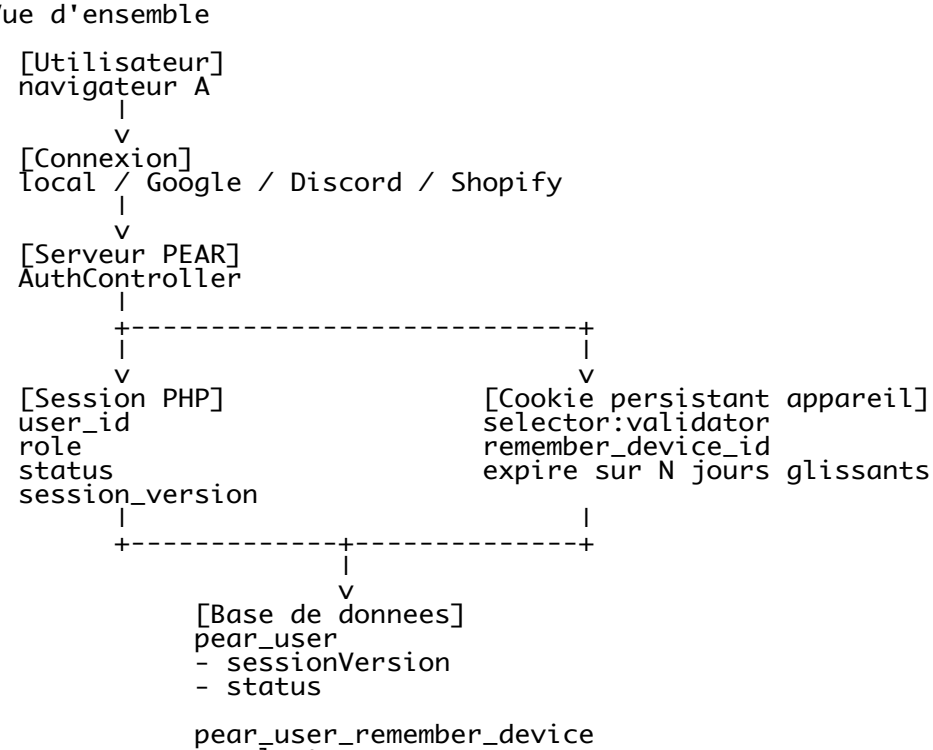
PEAR Collections - Schema fonctionnel metier
==========

Objectif
- Comprendre comment la connexion persistante, les appareils et la deconnexion
  globale se combinent du point de vue utilisateur / produit.

Vue d'ensemble



Validation a chaque page
- La session PHP doit exister
- La session doit contenir :
  - user_id
  - remember_device_id
  - session_version
- Le serveur relit pear_user.sessionVersion
- Le serveur verifie que le device existe encore
- Le serveur verifie que le device n'est pas expire

Si tout correspond
- La navigation continue
- Le device peut etre touche (lastUsedAt)
- Le token appareil peut etre tourne periodiquement

Si un controle echoue
- La session est detruite
- Le navigateur revient sur le formulaire d'authentification

Revoke globale / dure
- Compte utilisateur : "Deconnecter tous les appareils"
- Admin : /user/{id} -> "Deconnecter tous les appareils"
- Changement de mot de passe
- Changement d'email
- Bannissement

Effet attendu
- Les entrees pear_user_remember_device sont supprimees
- pear_user.sessionVersion est incrementee
- Toute autre session deja ouverte devient invalide
  au prochain chargement de page

PEAR Collections - Schema technique developpeur
==========

Objectif
- Montrer le flux technique entre session PHP, cookie appareil,
  pear_user_remember_device et pear_user.sessionVersion.

A. Login reussi (local / Google / Discord / Shopify)
  AuthController
  -> verify credentials / callback OAuth / login Shopify
  -> getUserBy... charge aussi sessionVersion
  -> seedSessionFromUser() stocke :
     user_id, login, pseudo, user_role, user_status, session_version
  -> RememberDeviceService.issueTokenForUser()
     cree / renouvelle pear_user_remember_device
     stocke remember_device_id en session
     ecrit le cookie persistant selector:validator

B. Requete suivante : validation centrale dans bootstrap
  config/bootstrap.php
  -> restoreSessionFromCookie() si pas de session active
     lit le cookie, retrouve selector, verifie hash + expiration,
     recharge l'utilisateur, recree la session, tourne le token
  -> validateActiveSession() a chaque requete
     recharge l'etat auth minimal, compare session_version,
     verifie remember_device_id et expiration du device

C. Controle dur de session active
  validateActiveSession() verifie :
  1. user existe encore
  2. user.status != banned
  3. $_SESSION['session_version'] == pear_user.sessionVersion
  4. $_SESSION['remember_device_id'] existe encore pour ce user
  5. le device n'est pas expire
  Sinon -> revokeCurrentDevice() + session_unset() + session_destroy()

D. Revoke globale / dure
  Db.revokeAllUserAuth(userId)
  1. DELETE FROM pear_user_remember_device WHERE userId = :userId
  2. UPDATE pear_user SET sessionVersion = sessionVersion + 1

  Appels prevus depuis :
  - AccountController : revoke all, password, email, delete account
  - ModerationController : bannissement, admin force logout all

Resultat attendu
- La session courante reste valide uniquement si elle est recreee proprement
- Les autres navigateurs ouverts tombent au prochain chargement
- Shopify est couvert car le flux d'auth final passe aussi par la creation
  d'une session utilisateur locale et d'un token appareil