Accounts in NoteDb

Edwin Kempin, Google
What is NoteDb?

- Store all Gerrit meta data in Git
- Get rid of the Database (ReviewDb)
Gerrit with ReviewDb vs. Gerrit with NoteDb
Why NoteDb?

● Enables atomicity
  (between branch state and review meta data)
● Consistent replication and backup
● Enables offline code review
● Enables code review federation between different servers
● No more database
  (less administration effort)
Gerrit Meta Data

- **Projects**
  (includes project config, access rights)
- **Changes**
  (includes change messages, comments, approvals, patch sets)
- **Accounts**
  (includes account properties, preferences, project watches, SSH keys, external IDs, stars, reviewed flags)
- **Groups**
  (includes memberships, audit logs)
<table>
<thead>
<tr>
<th>Projects</th>
<th>Already in Git since 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changes</td>
<td>Implementation is ready</td>
</tr>
<tr>
<td>Accounts</td>
<td>In Progress (Q4 2016)</td>
</tr>
<tr>
<td>Groups</td>
<td>Next (Q4 2016 / Q1 2017)</td>
</tr>
</tbody>
</table>
Account Data

- Account Properties
  (id, full name, preferred email, inactive)
- Preferences
  (general preferences, diff preferences, edit preferences)
- Project Watches
- SSH Keys
- Stars
- External-Ids
- Reviewed Flags
All-Users repository

Most account data is stored in a special All-Users repository

- One branch per user:
  refs/users/CD/ABCD
  (CD/ABCD is the sharded account ID)

- Updated through Gerrit REST API, but users may also fetch/push manually
All-Users repository

Most account data is stored in a special All-Users repository

- One branch per user:
  
  ```
  refs/users/CD/ABCD
  ```
  
  (CD/ABCD is the sharded account ID)

- Updated through Gerrit REST API, but users may also fetch/push manually

- Magic ref for own user branch
  
  ```
  refs/users/self
  ```
All-Users repository

- Users only have permissions for the own user branch
- Default access rights on
  
  \texttt{refs/users/\{shardeduserid\}}
Secondary Account Index

- Accessing the account data in Git is not fast enough for Account Queries.
- Implemented analogous to Change Index
- Enables arbitrary account queries
Account Data in User Branch

The user branch contains the following git config files:

- `account.config` (account properties)
- `preferences.config` (user preferences)
- `watch.config` (project watches)

In addition it contains an `authorized_keys` file with the SSH keys.
account.config

[account]
fullName = John Doe
preferredEmail = john.doe@example.com
active = false
• Default preferences can be configured in the `refs/users/default` branch.
[project "foo"]
notify = * [ALL_COMMENTS]
notify = branch:master [ALL_COMMENTS, NEW_PATCHSETS]
notify = branch:master owner:self [SUBMITTED_CHANGES]
authorized_keys

- Standard SSH file format
- Order defines sequence numbers
- Comment lines for deleted and invalid keys
External IDs

- Stored as Git Notes in All-Users:
  - Note Branch: refs/meta/external-ids
  - Note Key: SHA-1 of external ID
  - Note Content:
    ```
    [externalId "username:jdoe"]
    accountId = 1003407
    email = jdoe@example.com
    password = my-secret
    ```

- Ensures that External IDs are used only once.
- Use Account Index to lookup External IDs by account
Starred Changes

- Stored as refs in All-Users:
  
  refs/starred-changes/YY/XXXX/ZZZZZZZZZ

  (YY/XXXX = sharded Change ID, ZZZZZZZ = Account ID)

- Change Index contains Users that have starred a Change

- Support for labeled stars
Reviewed Flags

- Potentially thousands of Reviewed Flags per user, growing without bounds
- Storage in Git is bad
  (each update requires opening a repository and creating a commit)
- Stored in a local H2 database
- Extension Point to plugin other storage backends
  (e.g. required for multi-master)
Data Migrations

- Incremental
  (Migrate the data piece by piece whenever the storage format for a piece of the data is fully defined and implemented)
- Schema migrations on init
# Current State

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Index</td>
<td>Done</td>
</tr>
<tr>
<td>Account Properties</td>
<td>Open</td>
</tr>
<tr>
<td>Preferences</td>
<td>Done</td>
</tr>
<tr>
<td>Watched Projects</td>
<td>Implemented, Migration Pending</td>
</tr>
<tr>
<td>SSH Keys</td>
<td>Done</td>
</tr>
<tr>
<td>External IDs</td>
<td>In Progress</td>
</tr>
<tr>
<td>Starred Changes</td>
<td>Done</td>
</tr>
<tr>
<td>Reviewed Flags</td>
<td>Done</td>
</tr>
</tbody>
</table>
Thank You!

Questions?