

**GERRIT User Summit 2016**

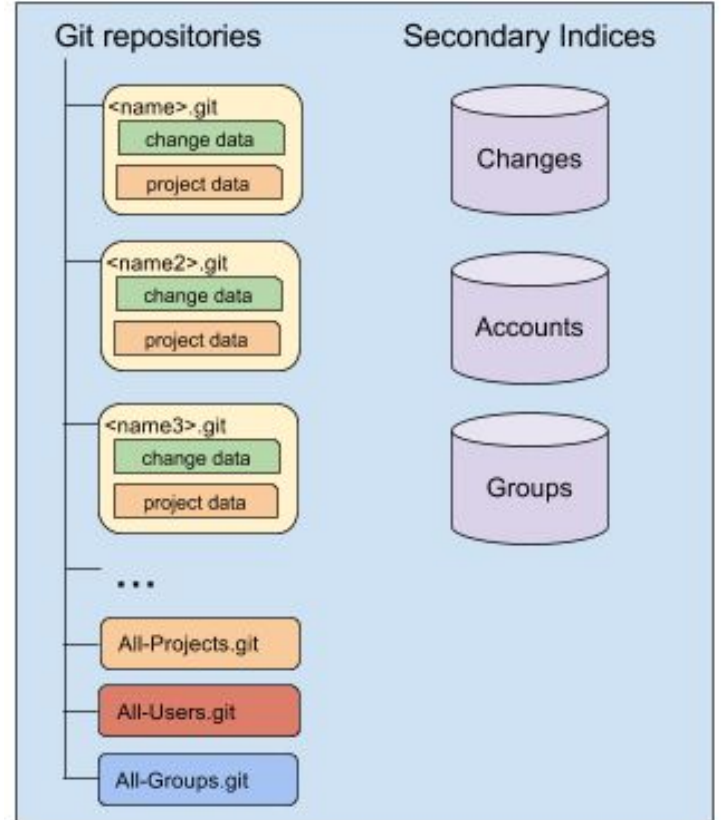
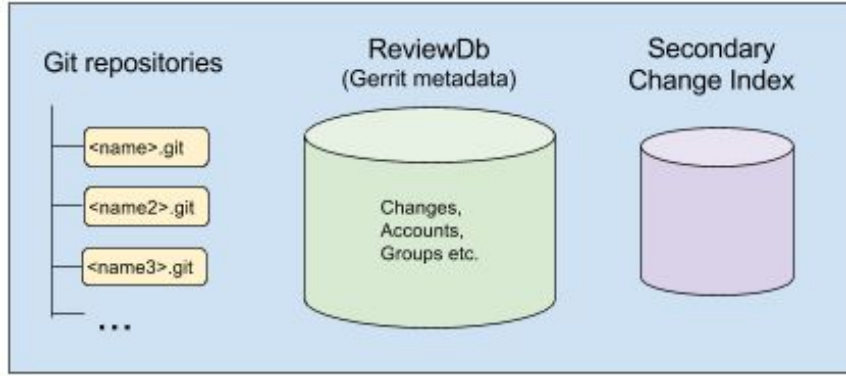
# Accounts in NoteDb

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)

# NoteDb - Current State

Projects	Already in Git since 2011
Changes	Implementation is ready
Accounts	In Progress (Q4 2016)
Groups	Next (Q4 2016 / Q1 2017)

# 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 `refs/users/${shardeduserid}`

<b>Reference:</b> <code>refs/users/\${shardeduserid}</code>	
<b>Read</b>	<input checked="" type="checkbox"/> Exclusive
<input type="button" value="ALLOW"/> Registered Users	
<b>Push</b>	<input checked="" type="checkbox"/> Exclusive
<input type="button" value="ALLOW"/> Registered Users	<input type="checkbox"/> Force Push
<b>Label Code-Review</b>	<input type="checkbox"/> Exclusive
<input type="button" value="-2"/> <input type="button" value="+2"/> Registered Users	
<b>Submit</b>	<input checked="" type="checkbox"/> Exclusive
<input type="button" value="ALLOW"/> Registered Users	

# 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
```

# preferences.config

```
[general]
  showSiteHeader = false
[diff]
  hideTopMenu = true
[edit]
  lineLength = 80
```

- Default preferences can be configured in the `refs/users/default` branch.

# watch.config

```
[project "foo"]  
  notify = * [ALL_COMMENTS]  
  notify = branch:master [ALL_COMMENTS, NEW_PATCHSETS]  
  notify = branch:master owner:self [SUBMITTED_CHANGES]
```

# authorized\_keys

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgQCgug5VyMXQGnem2H1KVC4/HcRcD4zzBqSuJBRWVonSSo
z3RoAZ7bWXCvVgWchtXwUURD689wFYdiPecOrWOUGeeyRq754YWRhU+W28vf8IZixgjCmiBhaL2gt3w
ff6pP+NXJpTSA4aeWE5DfNK5tZlxlSxqkKOS8JRSUeNQov5Tw== john.doe@example.com
# DELETED
# INVALID ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgQDm5yP7FmEoqzQRDyskX+9+N0q9GrvZeh5RG52EUpE4ms/
Ujm3ewV1LoGzc/IYKJAibdcZQNj9+06EfWZalRA3oOwAPe1eCnX+aLr8E6Tw2gDMQOGc5e9HfyXpC2p
DvzauoZNYqLALOG3y/1xjo7IH8GYRS2B7zO/Mf9DdCcCKSfw== john.doe@example.com
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQgQCaS7RHEcZ/zjl9hkWkqnm29Rnr2OQ/TZ5jk2qBVMH3BgZPsT
sEs+7ag9tfD8OCj+vOcwM626mQBZoR2e3niHa/9gnHBHFtOrGfzKbpRjTWtiOZbB9HF+rqMVD+Dawo/oic
X/dDg7VAgOFSPothe6RMhbgWf84UcK5aQd5eP5y+tQ== john.doe@example.com
```

- 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/ZZZZZZZ  
(YY/XXXX = sharded Change ID, ZZZZZZ = 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

Account Index	Done
Account Properties	Open
Preferences	Done
Watched Projects	Implemented, Migration Pending
SSH Keys	Done
External IDs	In Progress
Starred Changes	Done
Reviewed Flags	Done

**Thank You!**

**Questions ?**