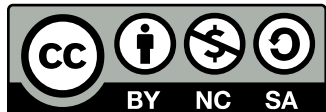


About Version Control API

Marco Villegas (marvil07)

<http://marvil07.net>

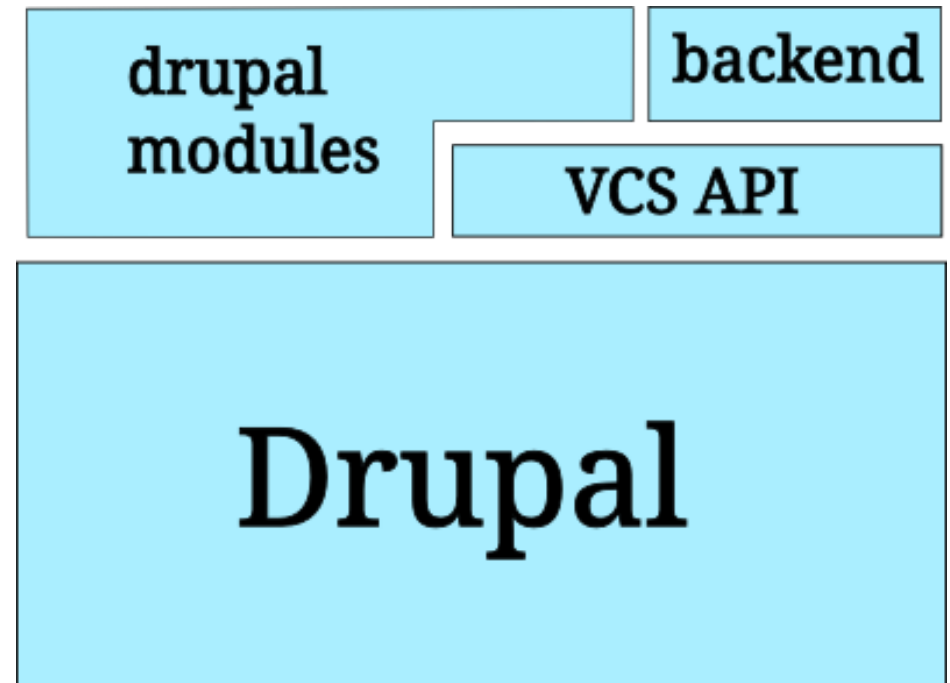
<http://drupal.org/user/132175>



This presentation is licensed under a
[Creative Commons Attribution-Noncommercial-Share Alike 3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Overview

- Authors
 - jpetso, sdboyer, hunmonk, marvil07
- Version Control Systems API
- Backends



Project page, wtf

Backends(2010-02-16)

- **Bazaar Backend**: no code, only placeholder.
- **CVS Backend**: versions for 5.x-1.x, 5.x-2.x and 6.x-1.x
- **Git backend**: versions for 5.x-1.x, 6.x-1.x and 6.x-2.x
- **Mercurial backend**: version for 5.x-1.x
- **Subversion backend**: versions for 5.x-1.x, 5.x-2.x and 6.x-1.x

About versions

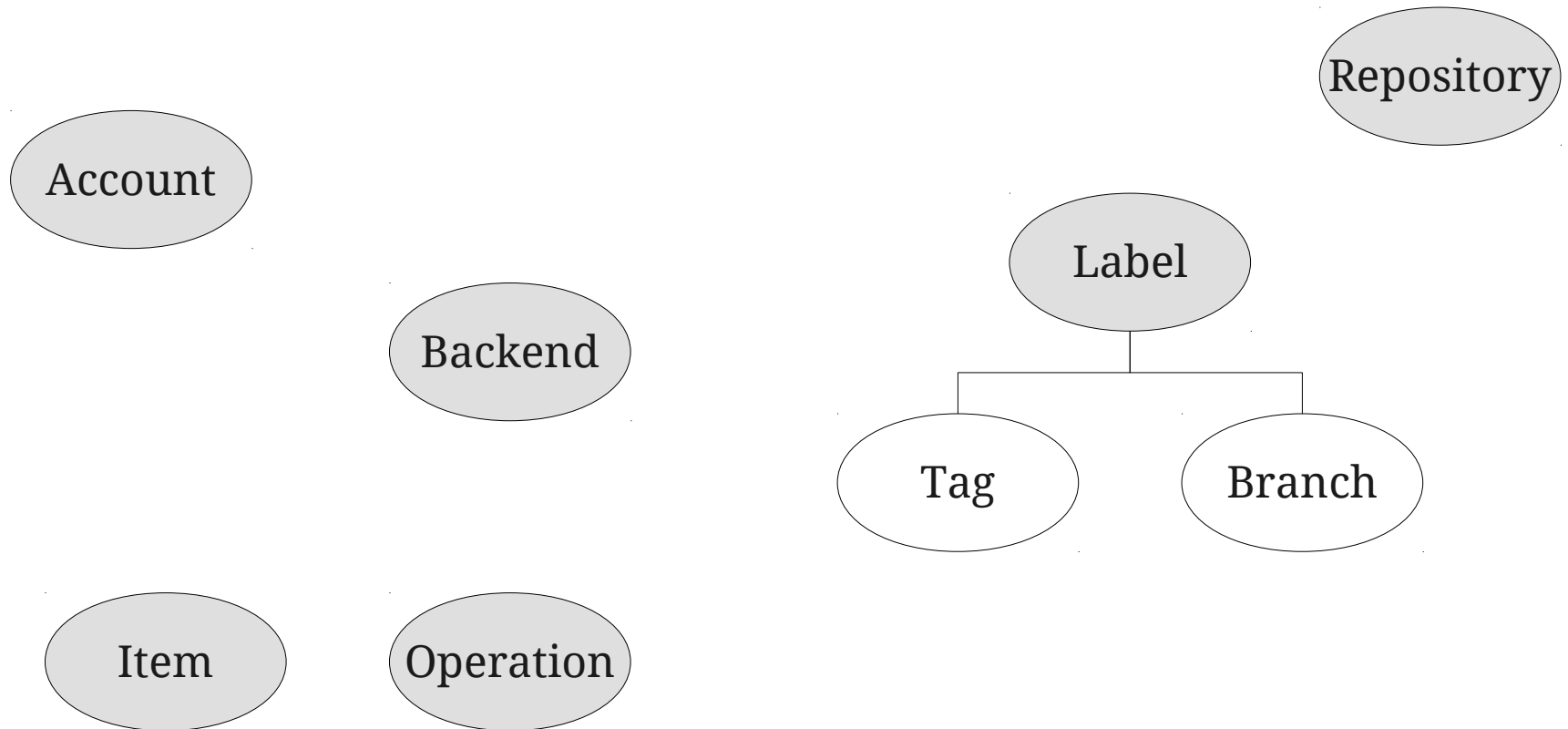
- 6.x-2.x-dev: HEAD now have 6.x-2.x, which is kind of a rewrite using OOP letting the modules to be easy readable ;-) **active development here**
- 6.x-1.0-rc2: the last 6.x-1.x release(it includes 5.x-2.0 changes)
- 5.x-2.0: fixed several architectural issues that came with 5.x-1.x. It's the last 5.x branch.
- 5.x-1.0: original API.

API changes

- **5.x-1.x -- 5.x-2.x**
- **5.x-2.x -- 6.x-1.x**
- 6.x-1.x -- 6.x-2.x(TODO: see **#606660: changelog from 6.x-1.x -- 6.x-2.x**)

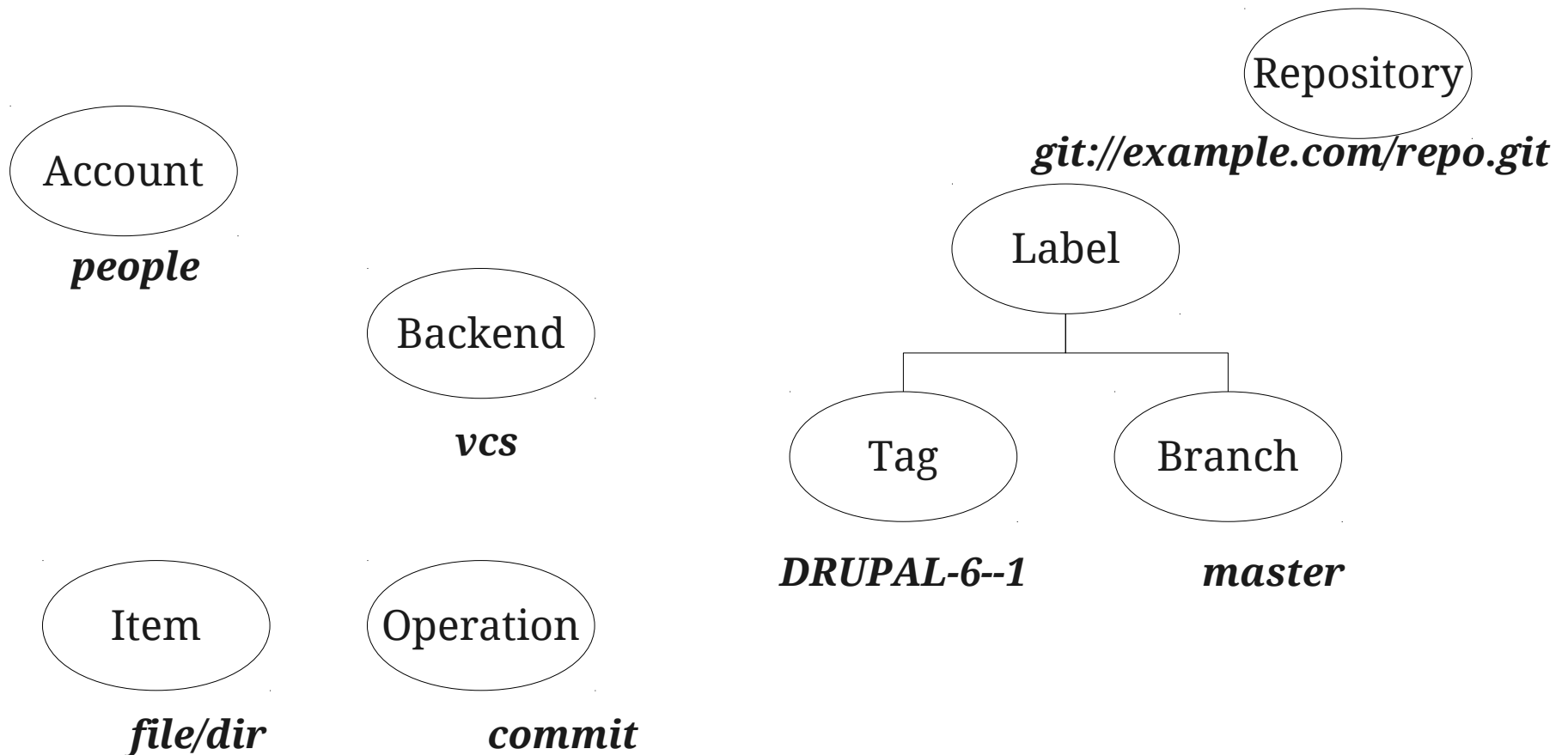
Abstraction: entities

- Versioncontrol<name> classes



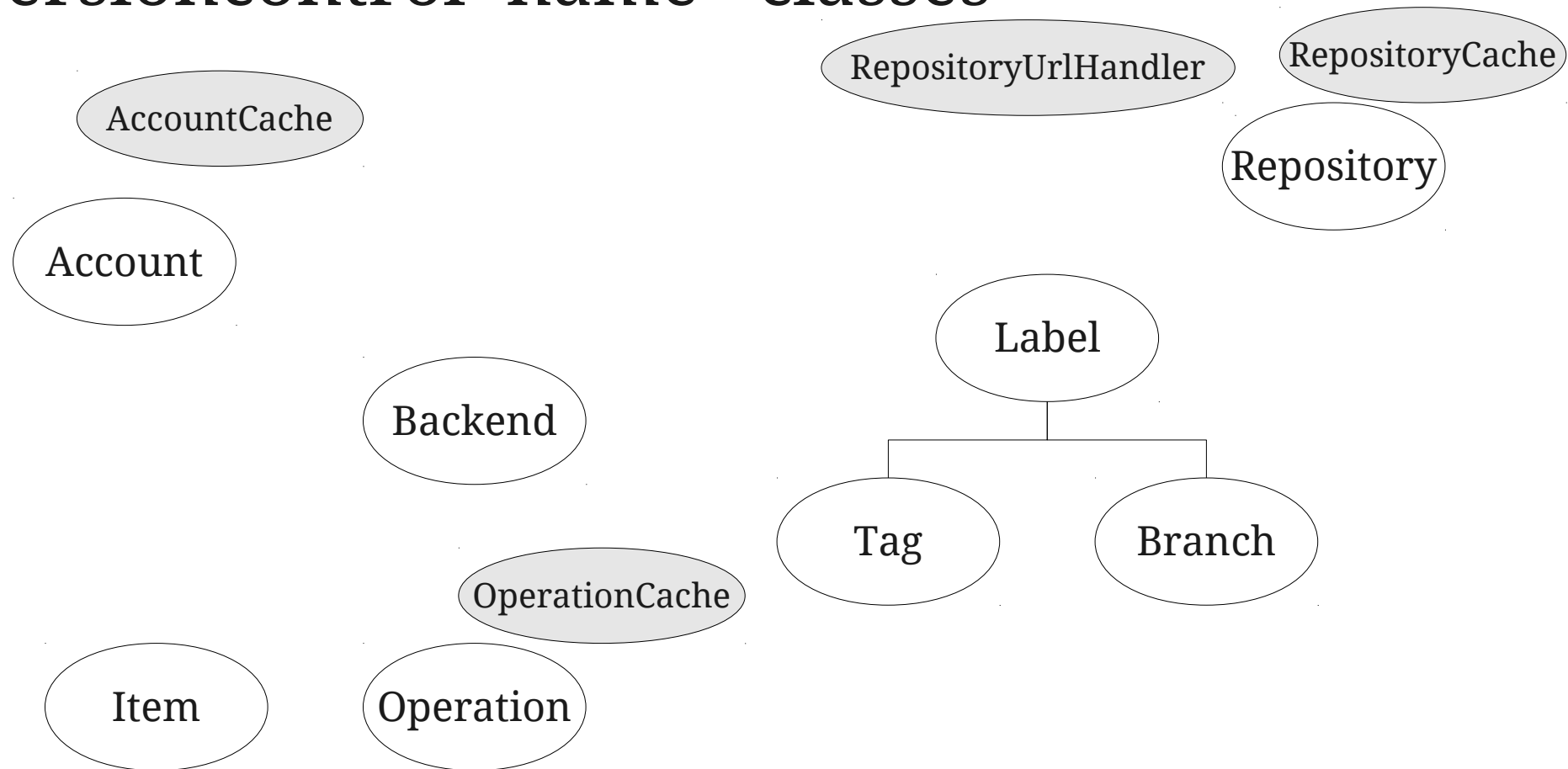
Abstraction: entities

- Versioncontrol<name> classes



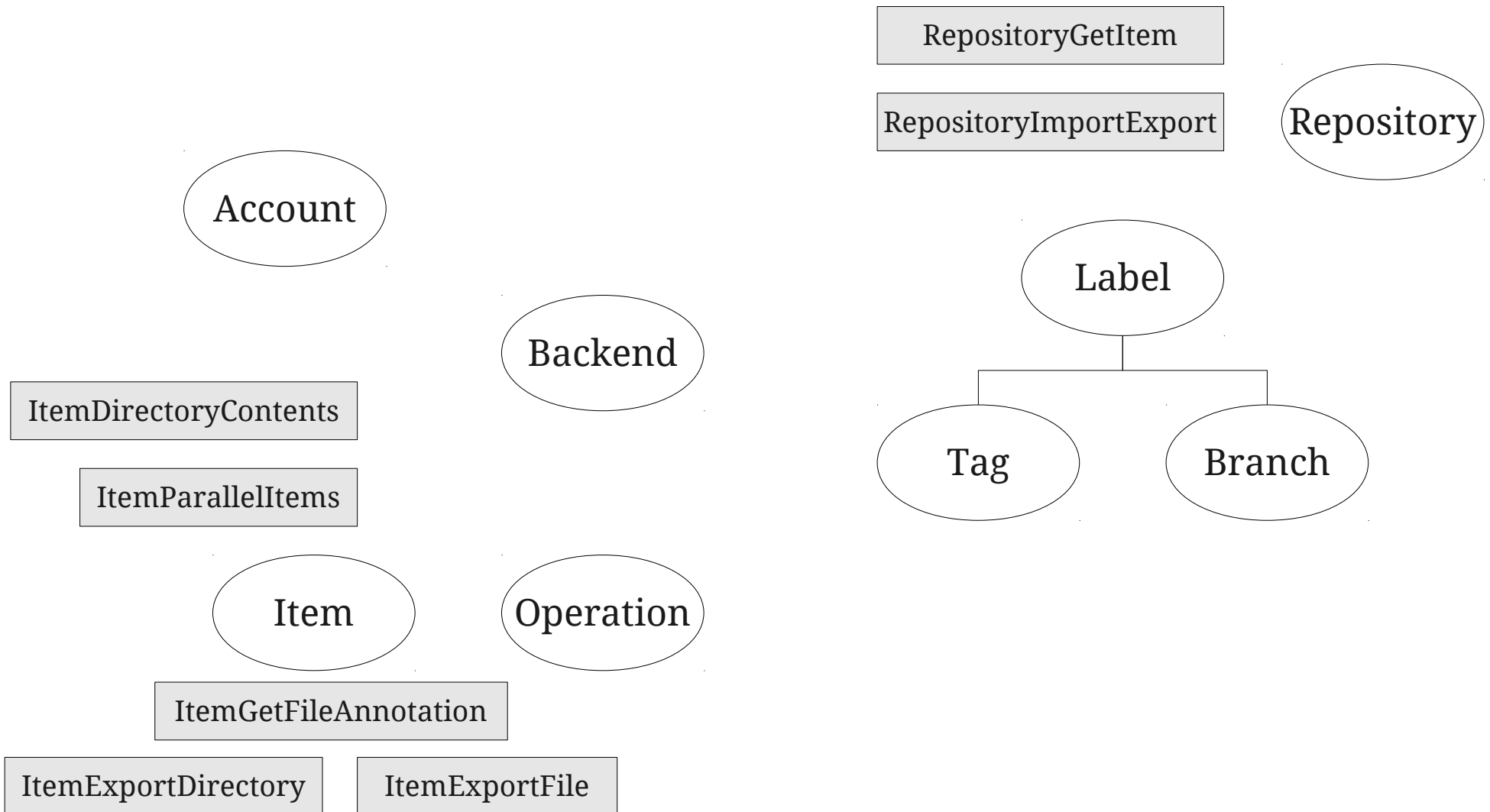
Abstraction: cache/url handler

- Versioncontrol<name> classes



Abstraction: interfaces

- Versioncontrol<name> classes



Really, what does it do?

- Commitlog
 - Show and filter logs from repos
 - Track code per user at account page
- Account status
 - Let users apply for a vcs account associated with a repo managing its status and optionally adding any agree checkboxes.
- Commit restrictions
 - Based on config per repo, using native pre-commit hooks of the VC, and if the backend implements it(aka backend need to support `VERSIONCONTROL_CAPABILITY_COMMIT_RESTRICTIONS` and/or `VERSIONCONTROL_CAPABILITY_BRANCH_TAG_RESTRICTIONS`).

But: what it can do?

- Backends suggestions:

```
public final function insert($additional_data = array()) {
    db_query(
        "INSERT INTO {versioncontrol_accounts} (uid, repo_id,
username)
        VALUES (%d, %d, '%s')", $this->uid, $this->repository-
>repo_id, $this->vcs_username
    );

    // Provide an opportunity for the backend to add its own
stuff.
    $this->_insert($additional_data);

    //..
}
```

But: what it **can** do?

- Backends suggestions:
 - `Operation::_insert($operation_items)`
 - `Operation::_delete($operation_items)`
 - `Repository::_getRepository()`
 - `Repository::_insert()`
 - `Repository::_update()`
 - `Repository::_delete()`
 - `Account::_update($additional_data)`
 - `Account::_insert($additional_data)`
 - `Account::_delete()`

But: what it can do?

- general hooks:
 - `versioncontrol_operation_labels($op, $operation, $labels)`
 - `$op`: insert, update, delete
 - `versioncontrol_operation($op, $operation, $operation_items)`
 - `$op`: insert, delete
 - `versioncontrol_repository($op, $repository)`
 - `$op`: insert, update, delete
 - `versioncontrol_account($op, $uid, $vcs_username, $repository, $additional_data)`
 - `$op`: insert, update, delete

Rules integration(almost[†])

- Token integration used at rules module
- Base API
 - Events
 - at operation insert
 - Conditions
 - Compare branch/tag name
 - Compare item paths
 - Compare repository
- commitlog
 - Actions
 - Send commit notification mail to: a registered user, an email address

[†] <http://drupal.org/node/788134>

The future

- Integrate commitlog to use views instead of own page
- Use dbtng (aka stop being our query builder)
- Implement batch api on *fetch now* link
- Issue queue:
 - <http://drupal.org/project/issues/versioncontrol>