Staging with git & drush

Markus Heurung – @muhh

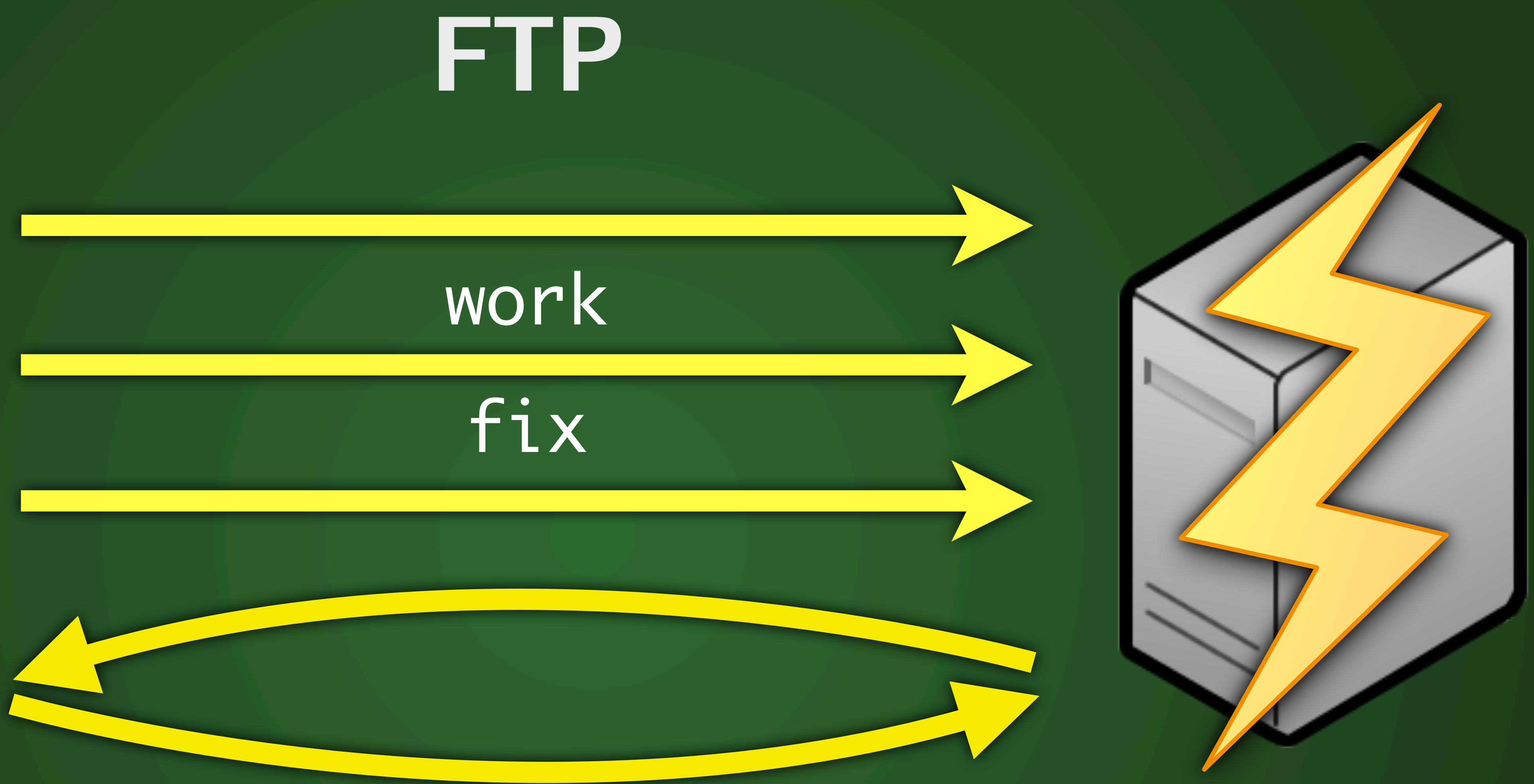
markus@freistil.it

freistil IT

DrupalCONCEPT

„Deploying“ the old way

```
index.php
update.php
modules/
includes/
sites/
…
```

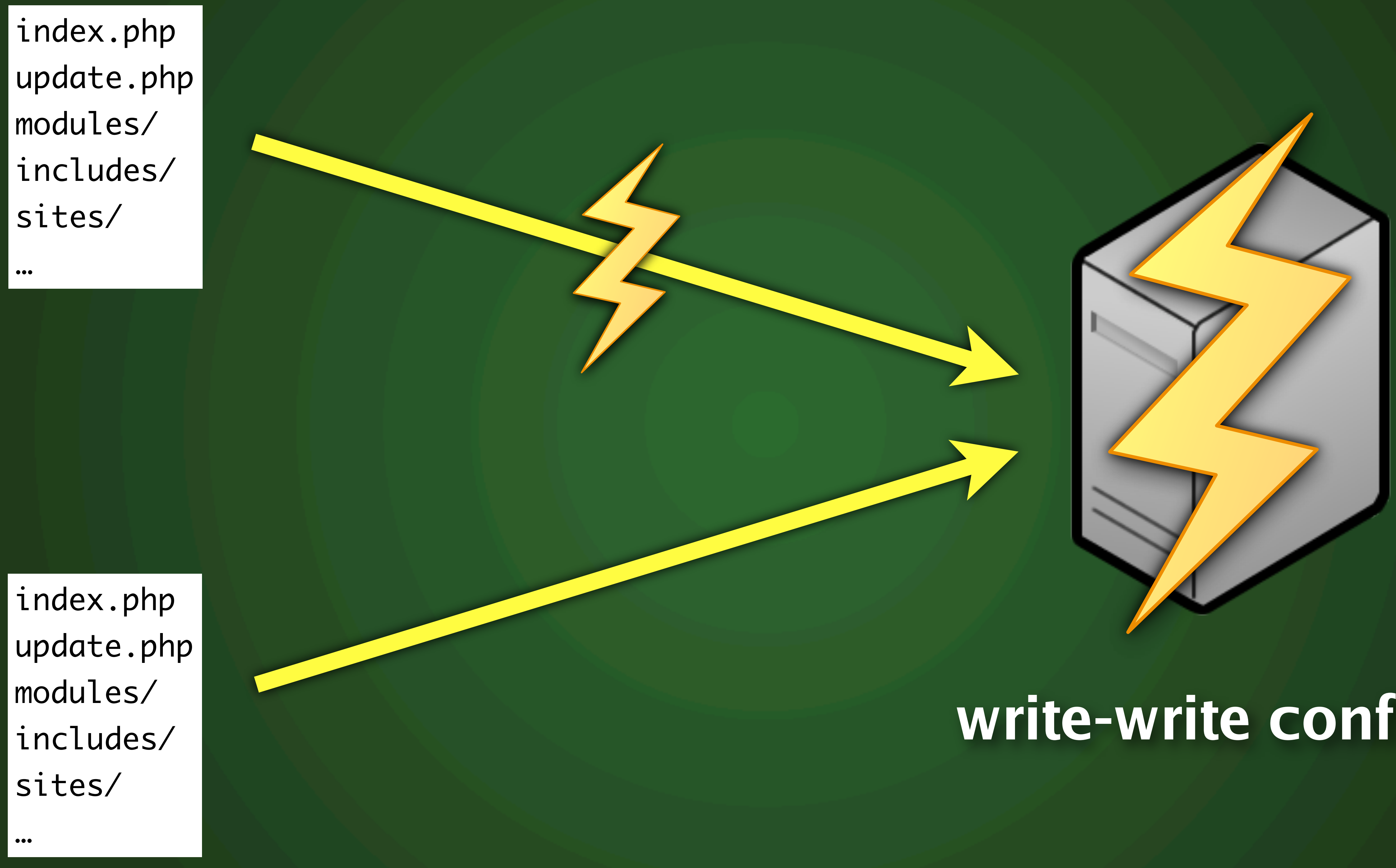


FTP

work

fix

multiple developers



```
index.php
update.php
modules/
includes/
sites/
…
```

```
index.php
update.php
modules/
includes/
sites/
…
```

write-write conflict

SUCKS!

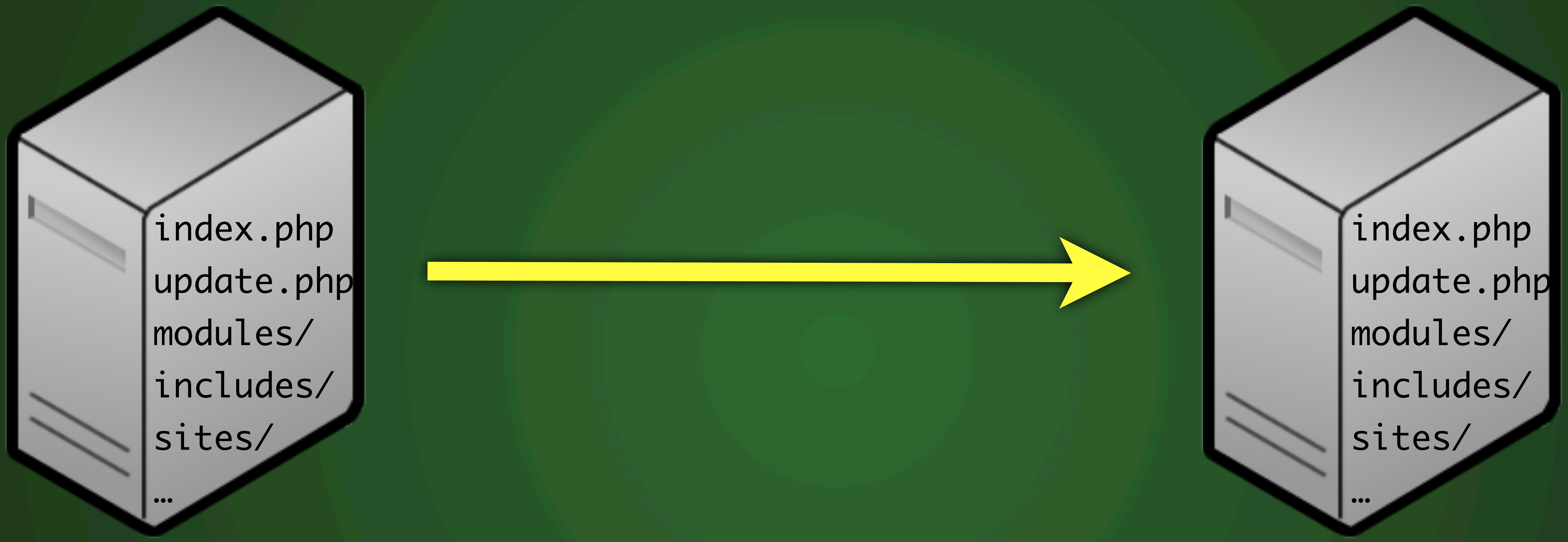
develop locally!

XAMPP/MAMP

Acquia Dev Desktop

Linux

Mac OS X



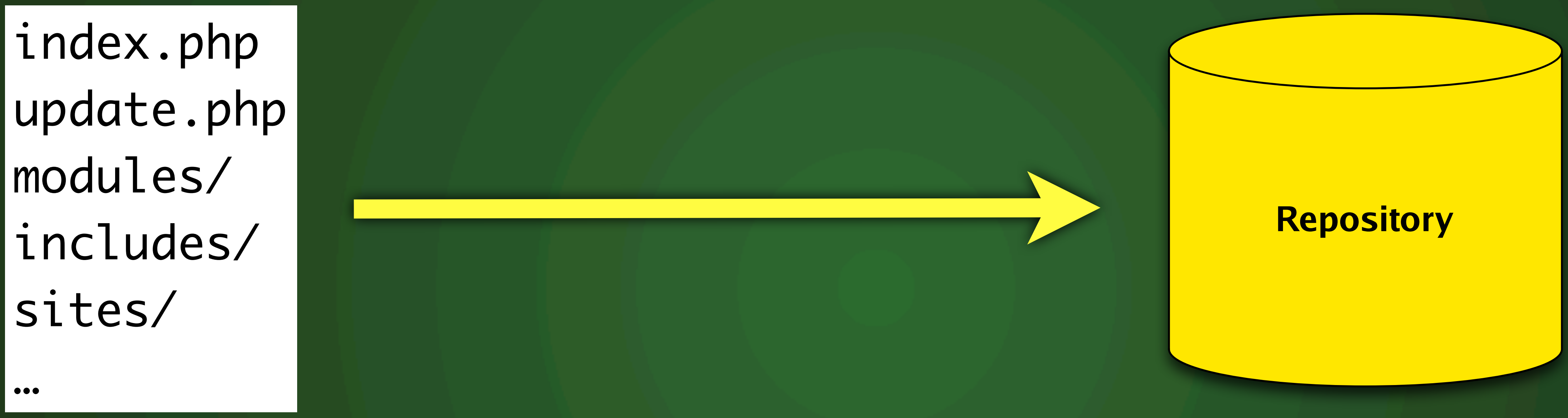
index.php
update.php
modules/
includes/
sites/
…

index.php
update.php
modules/
includes/
sites/
…

Version Control

Git

put your code into git



```
index.php
update.php
modules/
includes/
sites/
...
```

Repository

Some git basics

```
$ git add $FILE(S)
$ git commit $FILE(S)
```

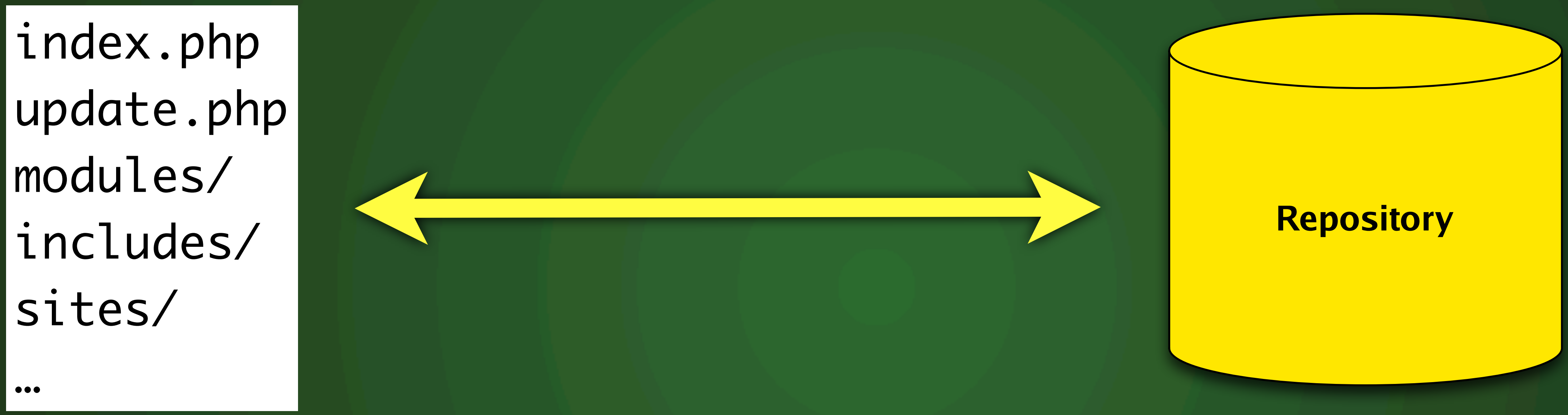
some git basics

```
$ git add docroot/
$ git commit docroot/
```

the typical local git workflow

```
$ [work …]
$ git commit -am "meaningful message"
$ [work …]
$ git add sites/all/modules/custom/stuff/
$ git commit -am "added stuff module"
```

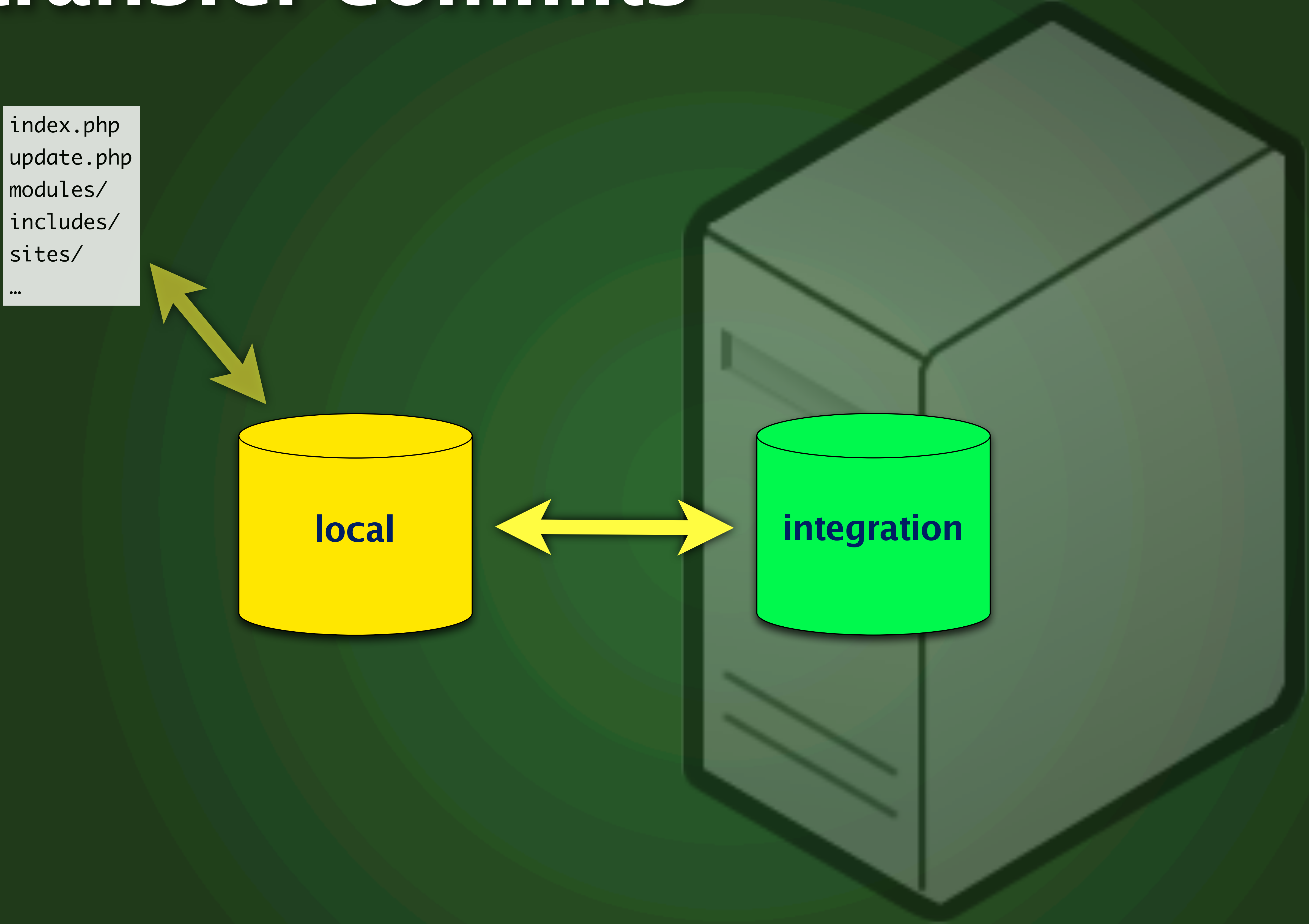
put your code into git



```
index.php
update.php
modules/
includes/
sites/
…
```

Repository

transfer commits



```
index.php
update.php
modules/
includes/
sites/
…
```

local

integration

git remote repositories

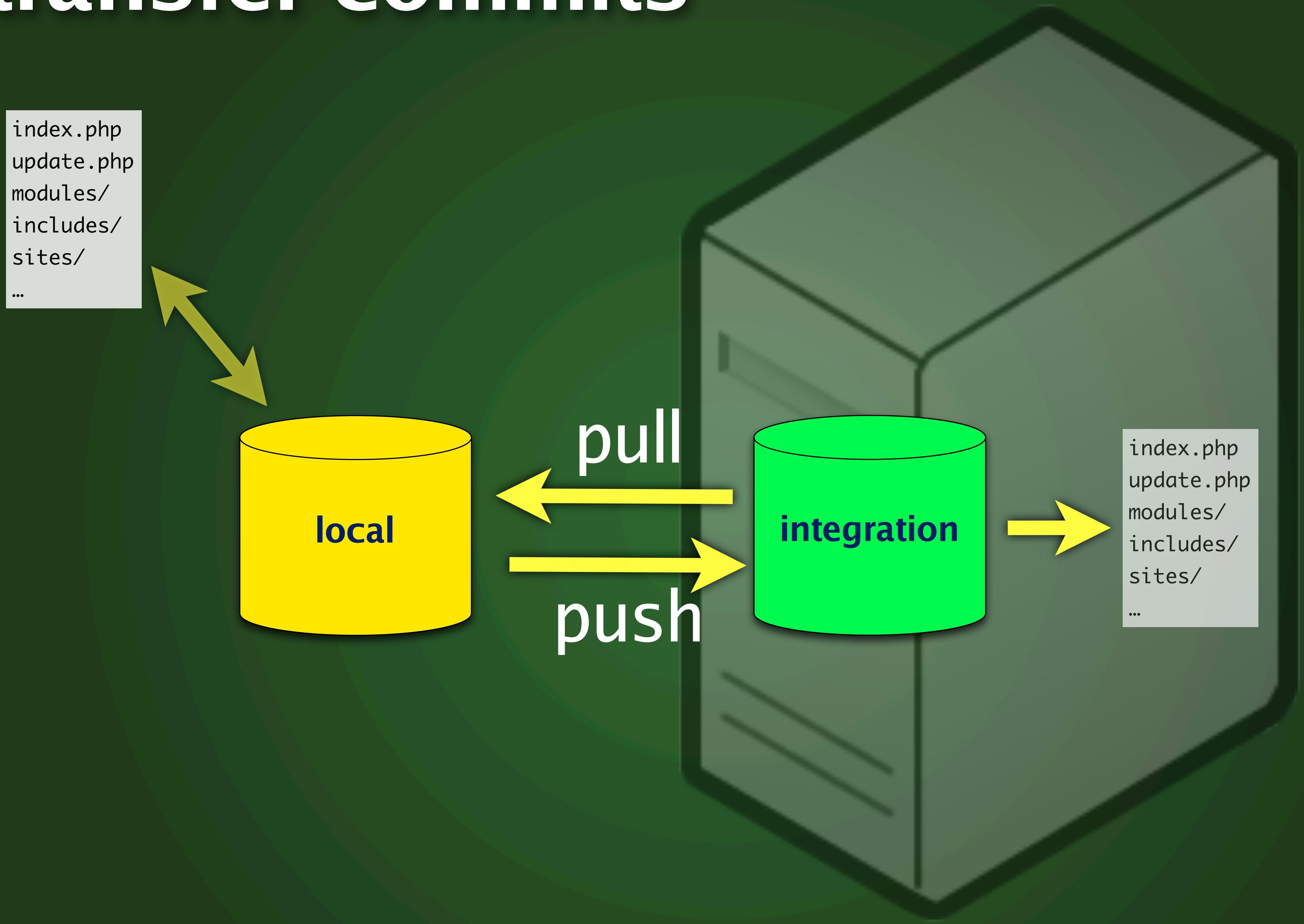
working with remotes

```
$ git remote add integration \\
  git@integration.server:test.git
```

transfer commits

```
$ git push integration
$ git pull integration
```

transfer commits



index.php
update.php
modules/
includes/
sites/
…

local

pull

push

integration

index.php
update.php
modules/
includes/
sites/
…

let the server deploy the code to its docroot

git knows hooks!

`.git/hooks/post-receive`

→ **go to docroot and do a git pull**

What about sites/*/files?

not in git!

let git ignore it

```
$ echo "sites/*/files" >> .gitignore
```

drush

some drush basics

```
$ drush pm-downlaod views
$ drush pm-enable views
$ drush pm-disable devel_themer
$ drush pm-update
$ drush updatedb
$ drush variable_set site_offline 1
```

and most used

```
$ drush cache-clear all
```

drush knows remotes, too!

called site-aliases

drush site-aliases

aliases.drushrc.php

```
$aliases['integration'] = array(
  'uri' => 'integration.server',
  'root' => '/var/www/integration.server/docroot',
  'remote-host' => 'integration.server',
  'remote-user' => 'integration-user'
);
```

drush site-aliases

```
$ drush @integration status
```

syncing files directory

use drush to sync files

```
$ drush rsync \\
   default:%files @integration:%files
```

syncing the database

in $aliases['integration']

```
'databases' => array(
  'default' => array(
    'default' => array(
      'driver' => 'mysql',
      'database' => 'integration',
      'username' => 'integration',
      'password' => 'supersecret',
      'host' => 'localhost',
      'prefix' => '',
      'collation' => 'utf8_general_ci',
    ),
  ),
),
```

in $aliases['integration']

```
'path-aliases' => array(
  '%dump-dir' => '/home/integration-user/db-dumps',
),
'command-specific' => array(
  'sql-sync' => array (
    'no-cache' => TRUE,
    'sanitize' => TRUE,
    'structure-tables' => array(
       'common' => array('cache', 'cache_menu', '…',
               'sessions', 'watchdog'),
),
```

sync your database

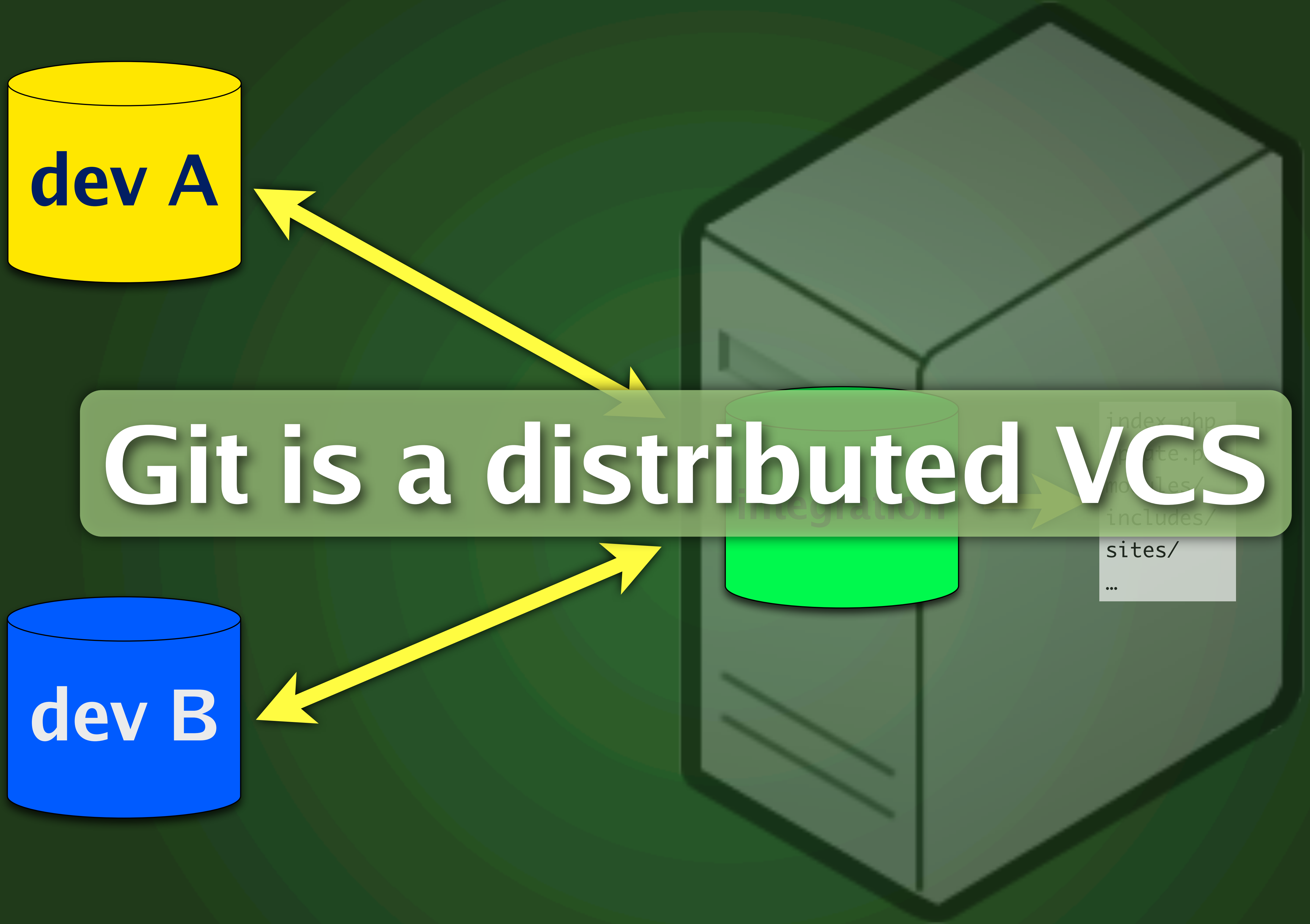
```
$ drush sql-sync \\
    default @integration
```

next problem:

multiple developers

no problem

Git is a distributed VCS

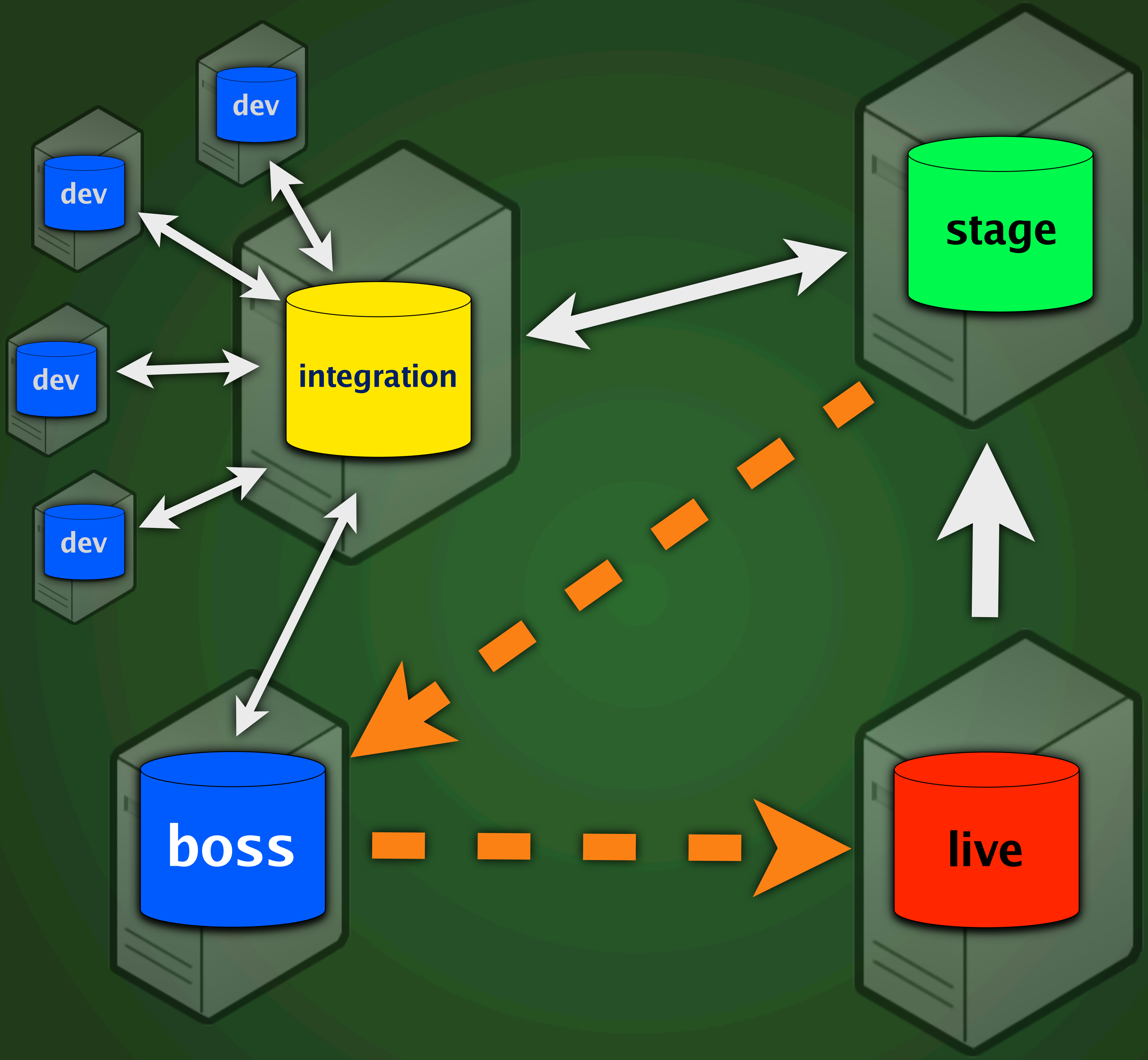


summary

- no forgotten files
- much faster uploads
- version history
- teamwork
- deployment to docroot on the server
- put as much in code as possible
  - features, strongarm, install profiles, ...
  - hook_update_N

code staging

a possible workflow


dev
dev
dev
dev
integration
stage
boss
live

Links

Git

http://git-scm.com
http://gitref.org/index.html
http://rogerdudler.github.com/git-guide/
http://sitaramc.github.com/gitolite/

Drush

http://drush.org
http://drupal.org/documentation/modules/drush

Questions!

Thank you!