

I have a dream!

(a look at some of our developer tools)

Simon Wheatley, @simonwheatley

© Camdiluv ♥ - <https://www.flickr.com/photos/camdiluv/4441155157/in/photostream/>

CODE
FOR THE PEOPLE



I'm Simon Wheatley, and I've been shouting incoherently from the back of the room.



Working with WordPress since 15 December 2005.



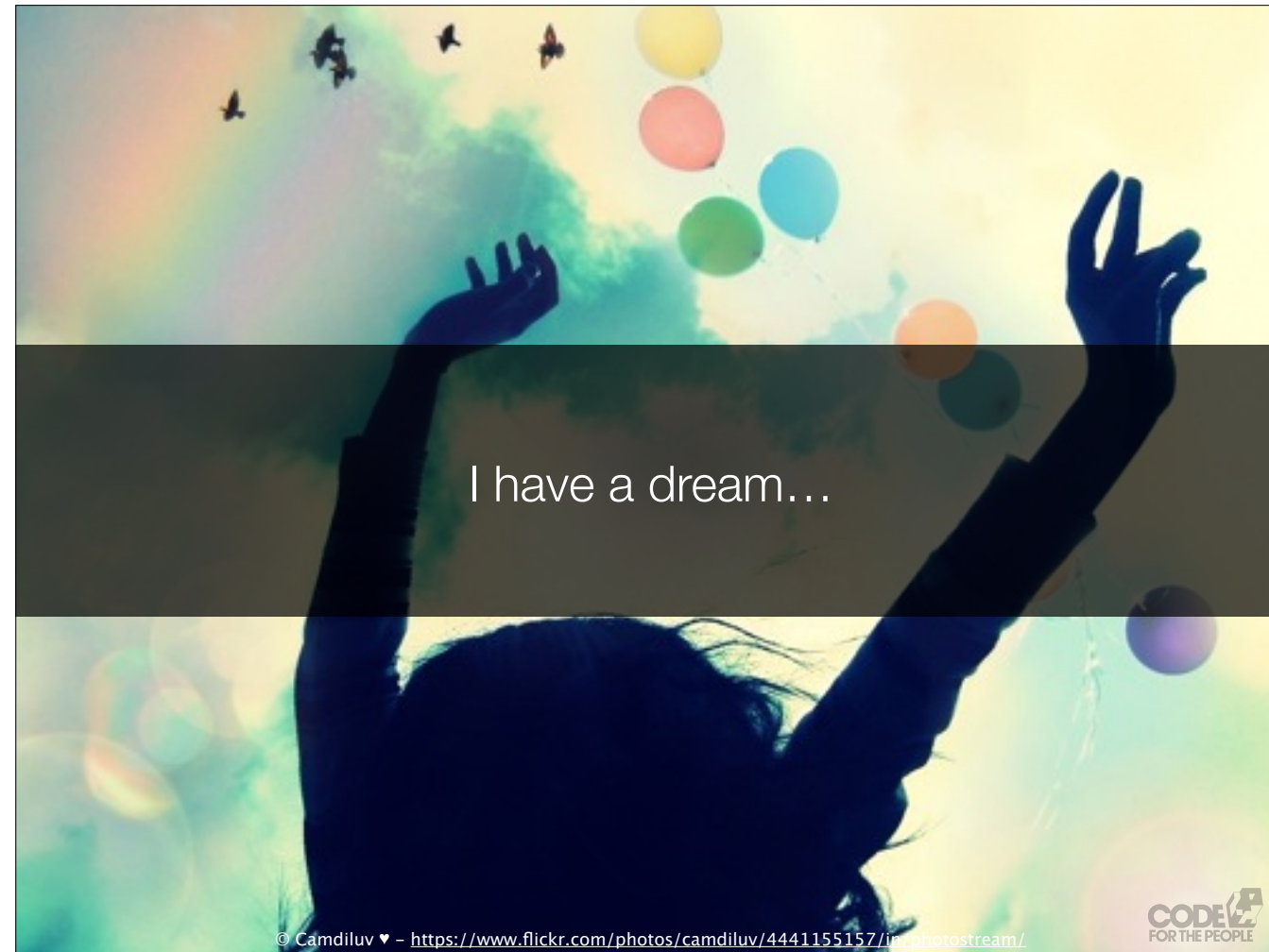
Founded on 25 January 2012, on the anniversary of the day Mike left the fateful comment on Matt's blog





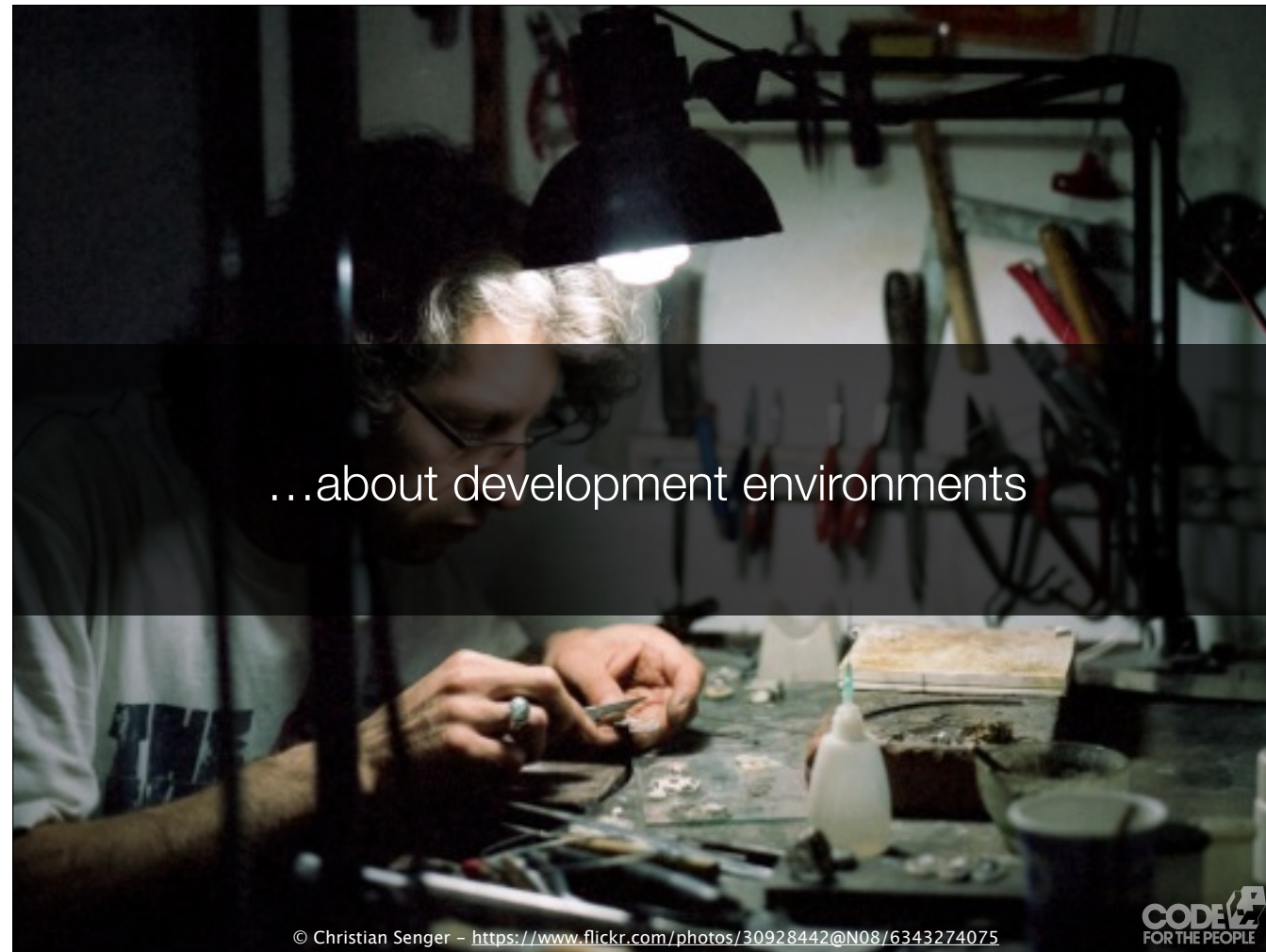
I work with Stephen Fry on all of his WordPress development.

IPC | MEDIA



© Camdiluv ♥ - <https://www.flickr.com/photos/camdiluv/4441155157/in/photostream/>

CODE
FOR THE PEOPLE



I want a common shared development environment for our team.

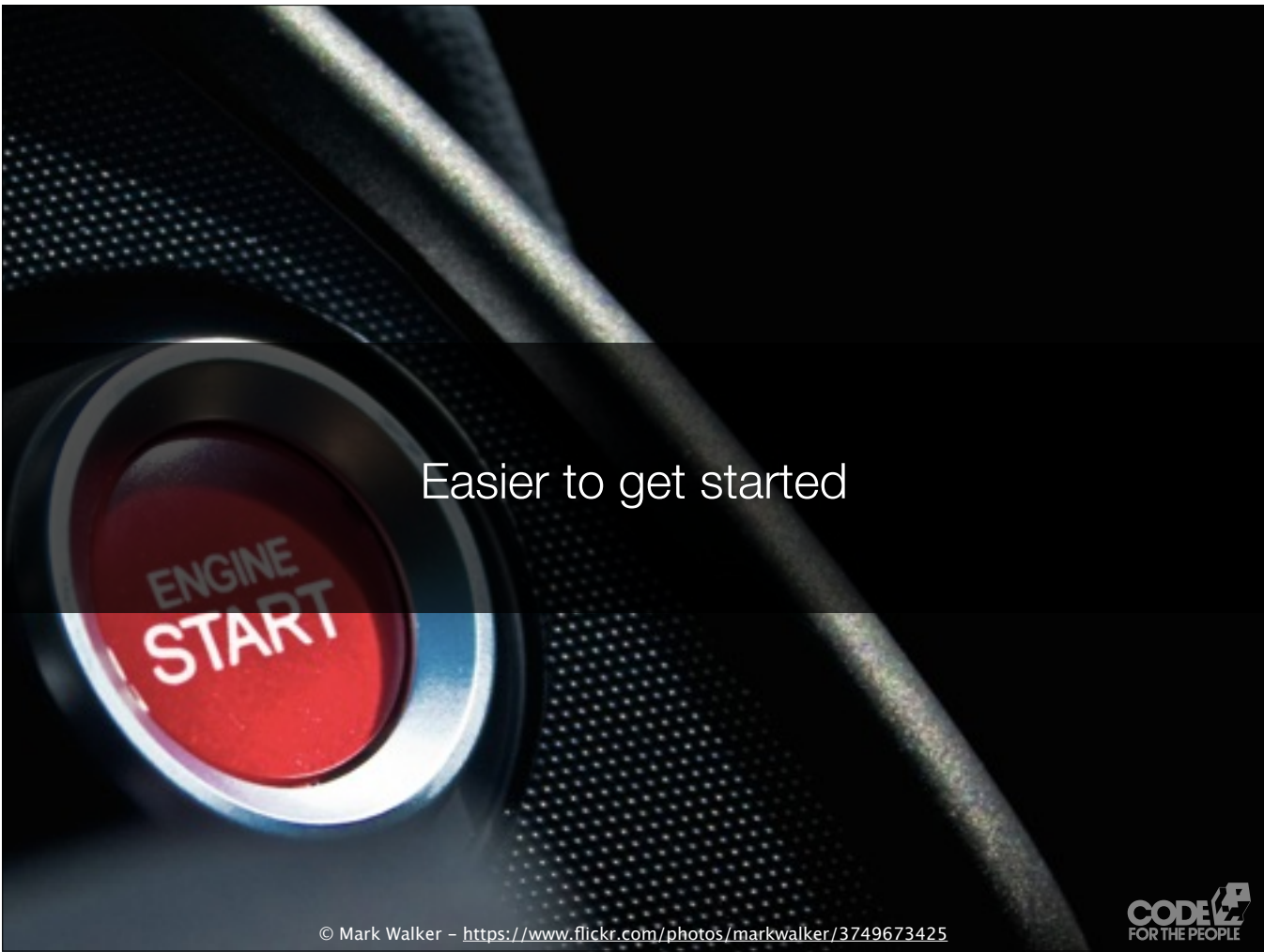
In this talk we're going to cover:

- * How we run the web servers for us to develop and test
- * How we set up projects on the development web server
- * How we manage and access the code
- * How we deploy our changes

Very fast skim. We'll dip into a bit of tech and code, but not much.

- * Vagrant
- * VVV
- * Bashscript
- * Public private key encryption
- * Git
- * Composer

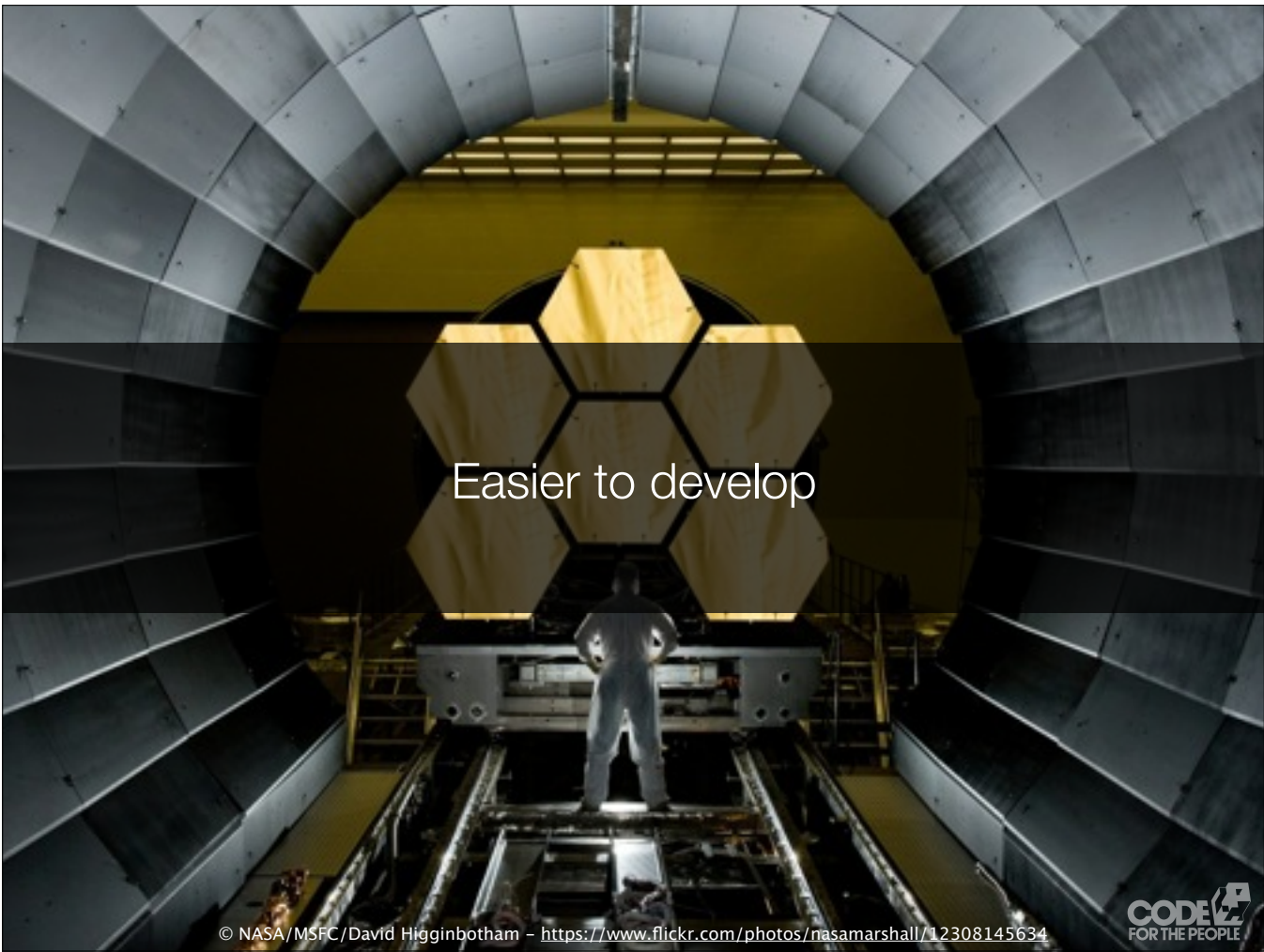
So why are we doing this?



Easier to get started

© Mark Walker - <https://www.flickr.com/photos/markwalker/3749673425>

CODE
FOR THE PEOPLE



Easier to develop

© NASA/MSFC/David Higginbotham - <https://www.flickr.com/photos/nasamarshall/12308145634>

CODE 12
FOR THE PEOPLE



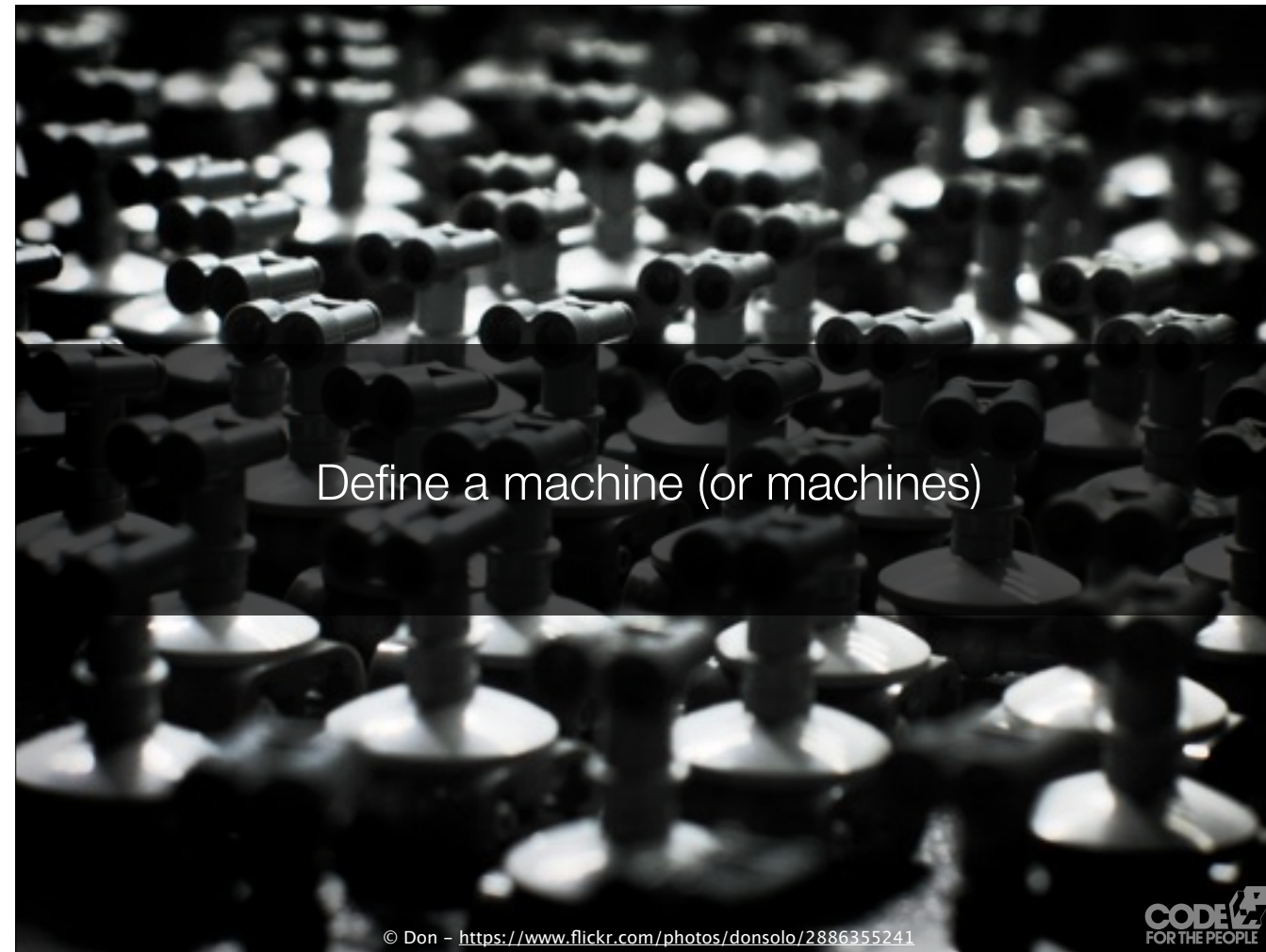
Easier to deploy

© Elle Jay Fisher – <https://www.flickr.com/photos/ellejayfish/2815231166>

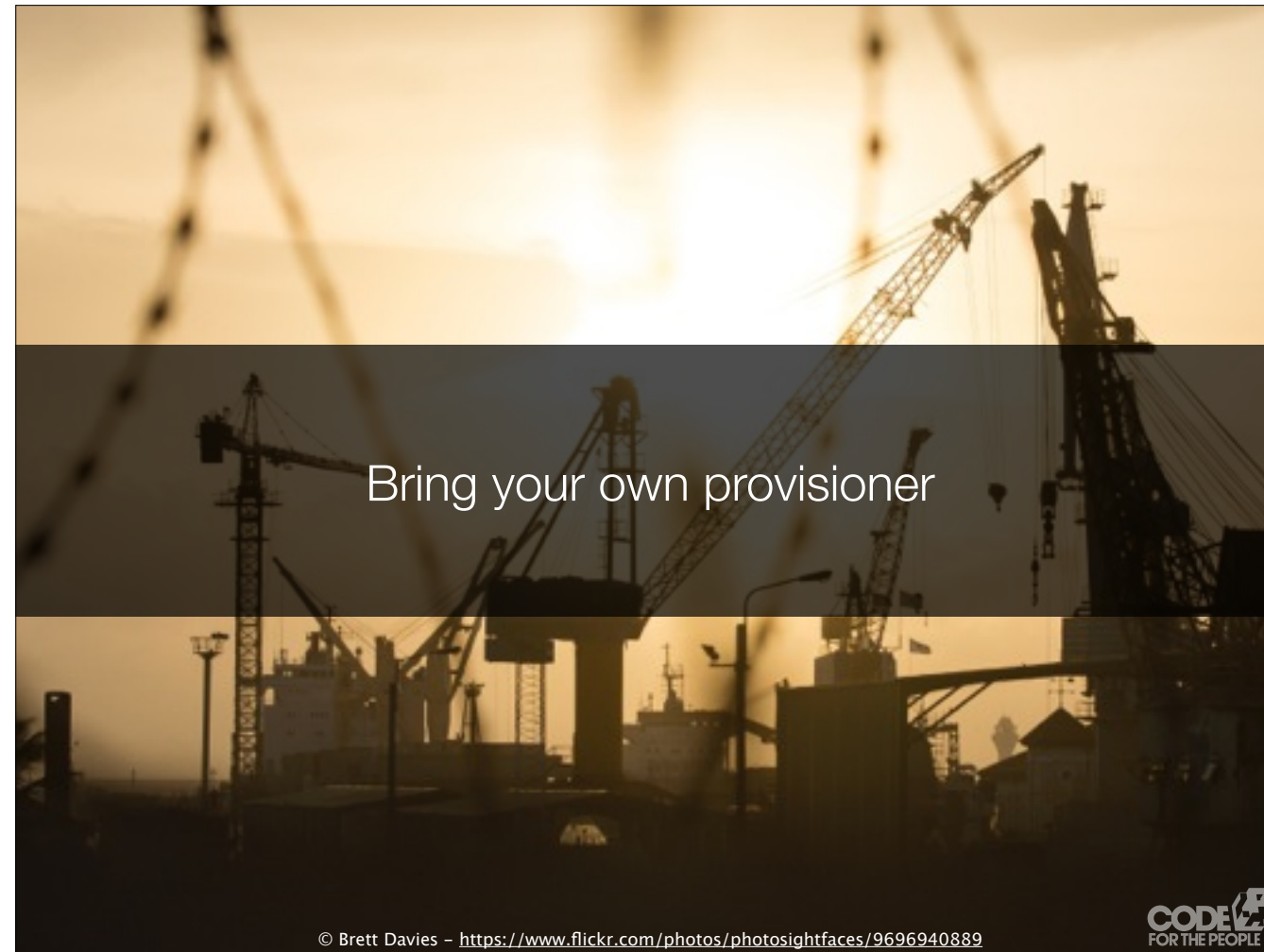
CODE4
FOR THE PEOPLE



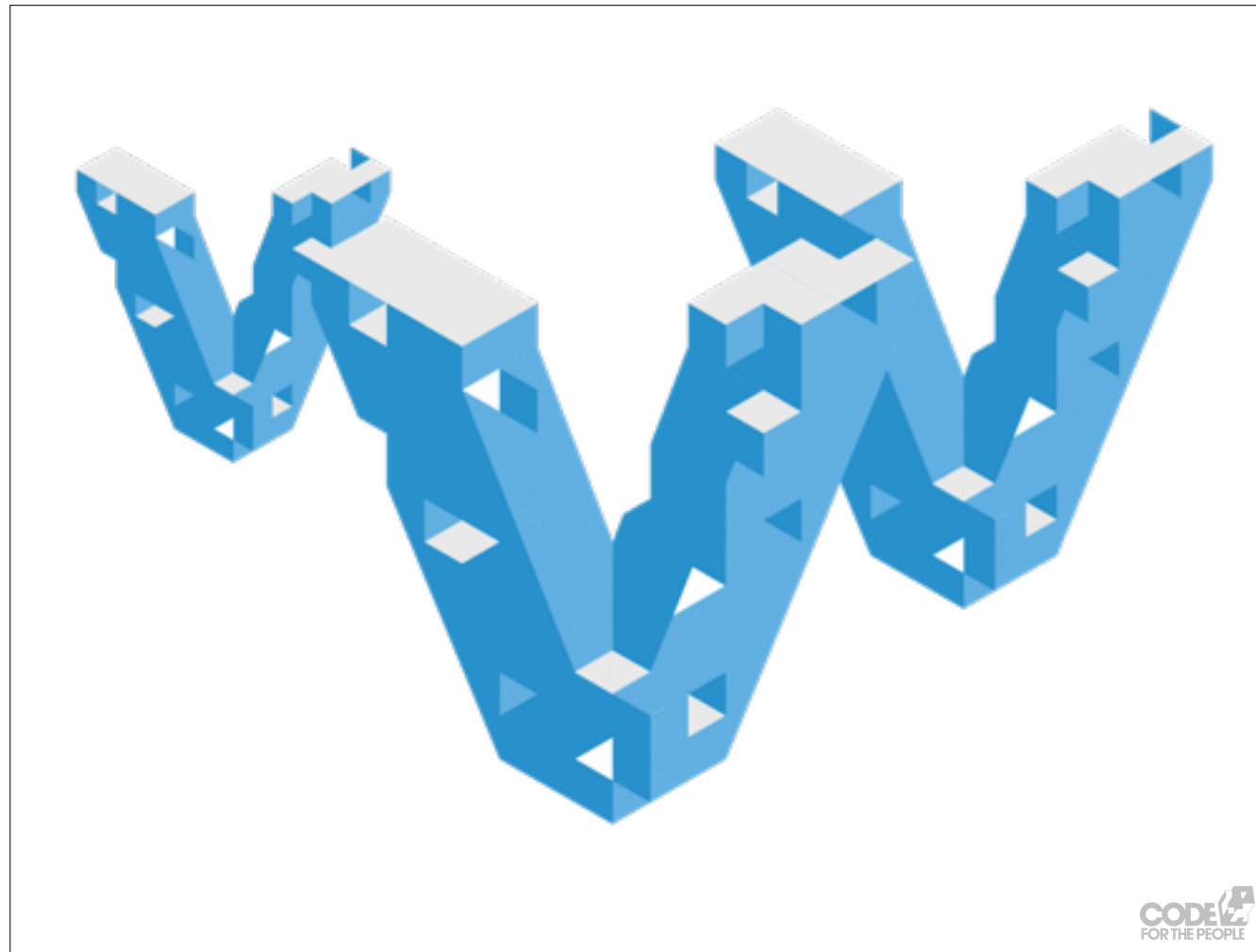
We base our development environments on Vagrant. Vagrant is...



...a tool for defining a virtual machine, or machines.



It allows you to use a variety of methods to provision that machine with everything you'll need, so you need a web server, you need PHP, and so on.



VVV, Varying Vagrant Vagrants, is a community project which originated with 10Up, an agency in the US. It's a WordPress specific Vagrant setup, which uses Bashscript to do the provisioning.



VVV gives us the Nginx web server...



CODE
FOR THE PEOPLE

...it gives us MySQL as a DB server...



CODE
FOR THE PEOPLE

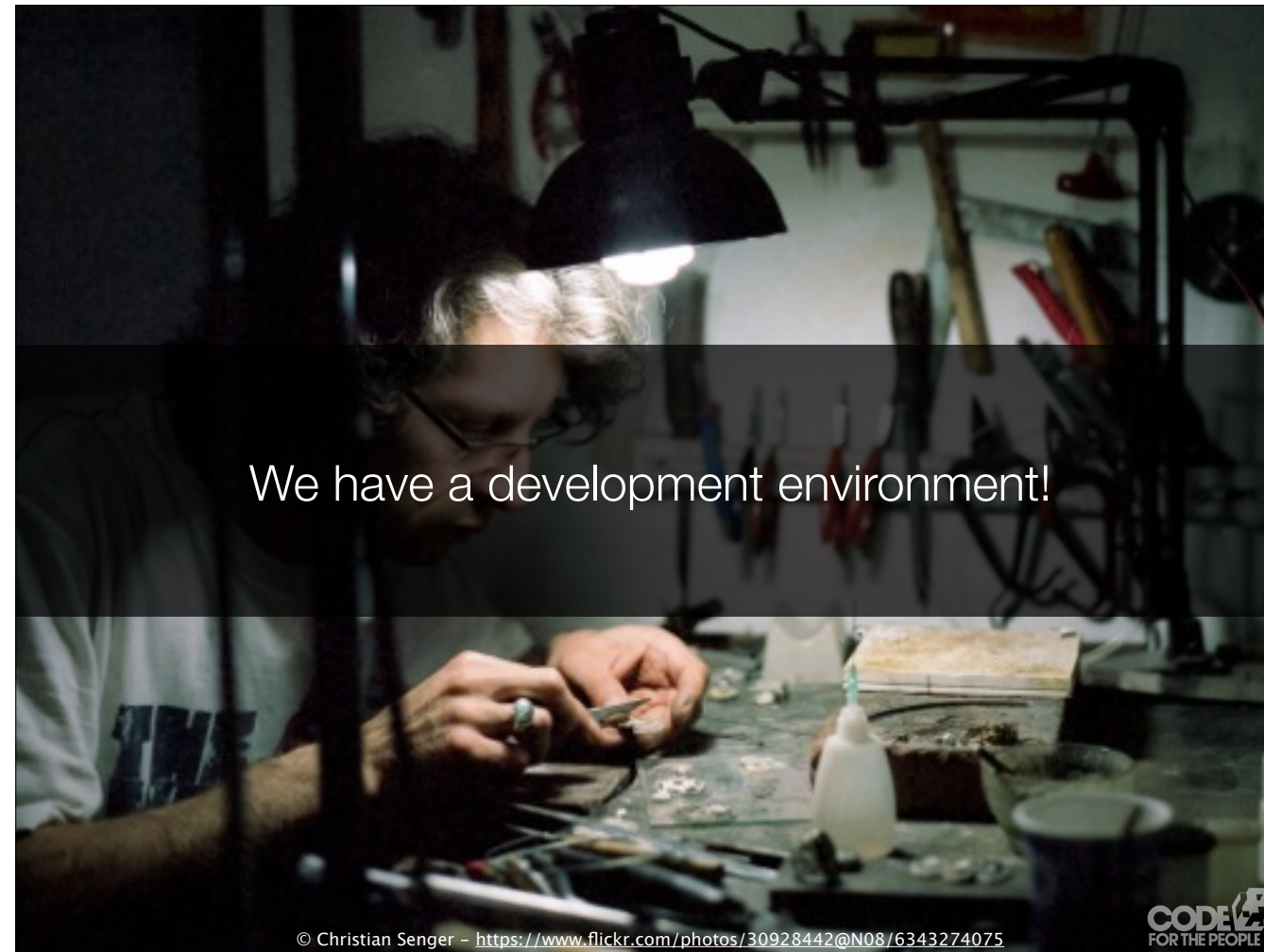
...it gives us PHP...

...and a ton more stuff under the hood, like PHPUnit for testing, code sniffers, tools for WordPress localisation, etc, etc.



Including some WordPress development sites for testing and if you want to help with Core WordPress development (which we do, at CFTP).

So if you're developing a plugin or a theme, you can just load up your plugin or theme into one of the WordPress development sites and go.



We've arrived! We have a development environment!



It's party time, let's get working...



...well, not quite. For a Project, rather than plugin or core development, we now need to set up some more things.



So what does a project look like?

We're going to need the web server to know about the project site we're developing. We're going to need a domain name we can type into a browser to see the site as we develop it. We're going to need the files for the project. We're going to need a database.



For this, for setting up a project site, we use VVV's Auto Site Setup, which was created by Weston Ruter, of X-Team, Jeremy Felt, who leads VVV, and myself.

To use it, you create a folder in the www directory of your Vagrant, which has three files in.

vvv-hosts

The first, vvv-hosts, defines the domain names.



vvv-nginx.conf



The second, vvv-nginx.conf, tells the webserver, Nginx, about your site.



vvv-init.sh



The third, `vvv-init.sh`, is a script that you write and which you will use to pull in your code, plugins, themes, and so forth, so you can work on the site.

```
vvv/www/my-project/  
vvv/www/my-project/vvv-hosts  
vvv/www/my-project/vvv-nginx.conf  
vvv/www/my-project/vvv-init.sh
```



For each project, you're going to have a new folder in the www folder, which is within your VVV vagrant folder.

vww-hosts

```
# Add as many hostnames  
# as you need here  
site-name.dev  
subdomain.site-name.dev  
another.site-name.dev
```

These get put into both your real and virtual machine's `/etc/hosts` files. `/etc/hosts` is a way of faking DNS, so your web browser knows which server to talk to about your project site.

vvv-nginx.conf



This one might sound scary, but it's not.

vvv-nginx.conf

```
server {  
    listen      80;  
    listen      443 ssl;  
    server_name site-name.dev;  
  
    root        {vvv_path_to_folder}/htdocs;  
  
    include /etc/nginx/nginx-wp-common.conf;  
}
```



Pretty simple template, this configures your project site on your VVV Nginx web server. The blue bit gets replaced

```
server {  
    listen      80;  
    listen      443 ssl;  
    server_name site-name.dev *.site-  
name.dev;  
  
    root        {vvv_path_to_folder}/htdocs;  
  
    include /etc/nginx/nginx-wp-common.conf;  
}
```



vvv-init.sh



vvv-init.sh is the heart of the system. This script is where you tell WordPress to assemble your project, get your code, create your database and so forth.

We'll walk through the various tasks that this init script will need to do.

So the first issue we are probably going to hit here, is...

<https://github.com/cftp/vvv-init/>

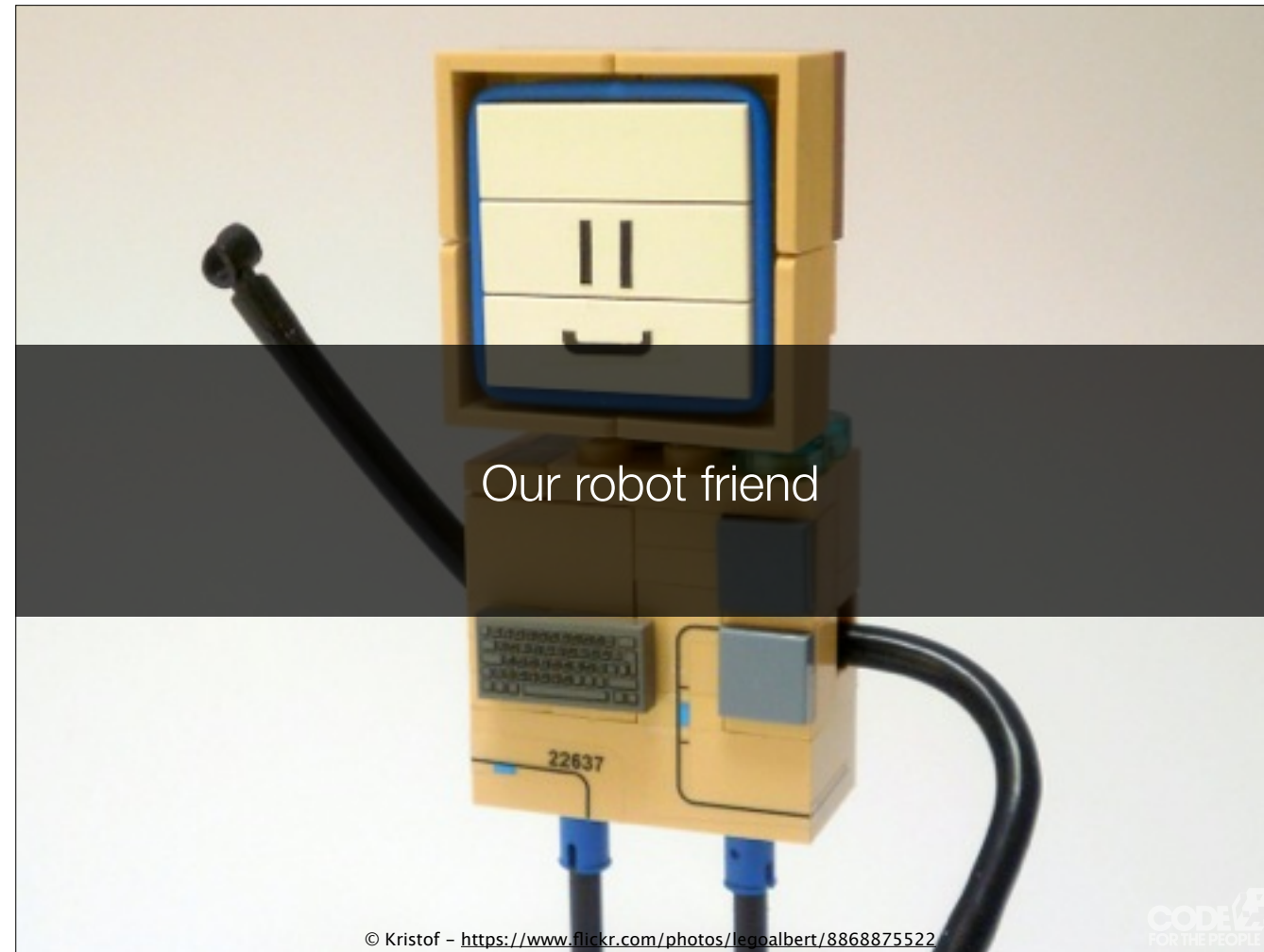


You can see the work in progress starter project on GitHub, that I use to setup a new project. It's lacking in documentation, but you might find it interesting. I'm also writing a blogpost series about how all this works, first post is published and the second is half written.



...is if we keep our code in private repositories, as we tend to for client projects.

The init script is running headless, meaning no user input, so we can't enter passwords and we can't answer any questions the script has during execution. This is a problem when we put our code in private repositories.



© Kristof – <https://www.flickr.com/photos/legoalbert/8868875522>

CODE
FOR THE PEOPLE

To solve this problem, we're going to create a robot friend.

He is going to have an account on all our private repositories on GitHub, GitLab, Bitbucket, etc. We care very little about his privacy, his login credentials are going to be distributed with every project, so it's very important that his account can NOT write to anything.

He's going to use public/private key encryption to log in to all our private code repositories and get stuff for us.

We're going to give his public AND private key to every developer working on the project.

Authentication



We need the script to authenticate as our robot friend.

Man in the middle attack!



We need the script to recognise servers even if it hasn't seen them before. Otherwise it's going to FREAK OUT. If you've used SSH, you'll remember the "MAN IN THE MIDDLE ATTACK" message you always have to clear through.


```
vvv-init.sh
# Prepopulating known good host signatures
mkdir -p ~/.ssh
touch ~/.ssh/known_hosts
IFS=$'\n'
for HOST in $(cat "ssh/known_hosts"); do
    # ...loop over and add the hosts if
    # not already present...
done

# http://tmx0009603586.com/help/en/entpradmin/Howto\_KHCreate.html
```



Load in the signatures of the servers we need to contact, so we will “know” them when we see them.

```
vvv-init.sh  
# Loading a key for a private repository  
  
ssh-agent bash -c \  
    "ssh-add ssh/cftp_deploy_id_rsa; \  
    git clone $REPO_SSH_URL htdocs;"
```



Now we can use ssh-agent to load up our robot friend's private key, so we can authenticate as him. Then pull down the Git repository.

```
vvv-init.sh  
# Create the database if it's not there  
  
mysql -u root --password=root \  
-e "CREATE DATABASE IF NOT EXISTS \  
$DB_NAME; GRANT ALL PRIVILEGES ON \  
$DB_NAME.* TO wp@localhost \  
IDENTIFIED BY 'wp';"
```



A quick command to create the database and give our WP user some credentials.

```
# Create WordPress config
```

```
if [ ! -f htdocs/wp-config.php ]; then  
    wp core config --dbname="$DB_NAME" \  
        --dbuser="wp" --dbpass="wp" \  
        --dbhost="localhost" \  
        --extra-php <<PHP  
$EXTRA_CONFIG  
PHP  
fi
```

```
vvv/www/my-project/  
vvv/www/my-project/htdocs/  
vvv/www/my-project/htdocs/(project stuff)  
vvv/www/my-project/vvv-hosts  
vvv/www/my-project/vvv-nginx.conf  
vvv/www/my-project/vvv-init.sh
```



Once this has all run, for our projects, we not have an htdocs folder which contains all the project files, WordPress, etc, etc.



So let's talk a bit about Composer.



Composer is a system for defining what you want in your project, using a (roughly) human readable JSON file. The dependencies can be pulled in as plugins from WordPress.org...



... or each as separate repositories. This is the bit that really works for us. So each of our plugins within each of our projects is a separate repository, and changes to one can be pulled into all.

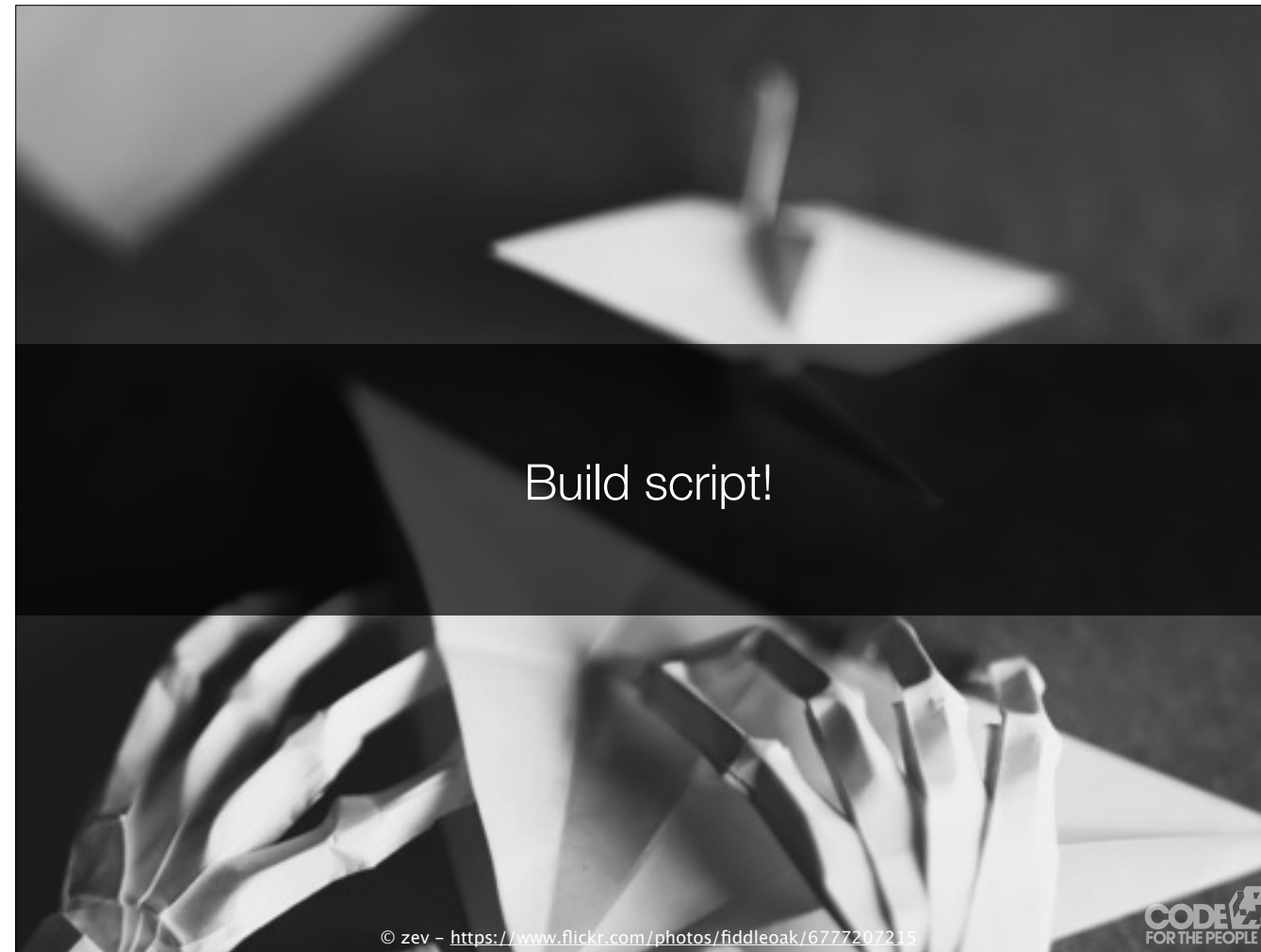
So if John is working on a plugin which handles organising content by popularity, that plugin can be easily pulled into multiple projects in a developer friendly format. In it's own Git repository. So when we fix a bug, or add functionality, on one project the changes can be pushed back to GitHub or wherever, and the other project can pull those changes in.



Our final step is deployment. Getting the site live, or pushing the changes to the live site.



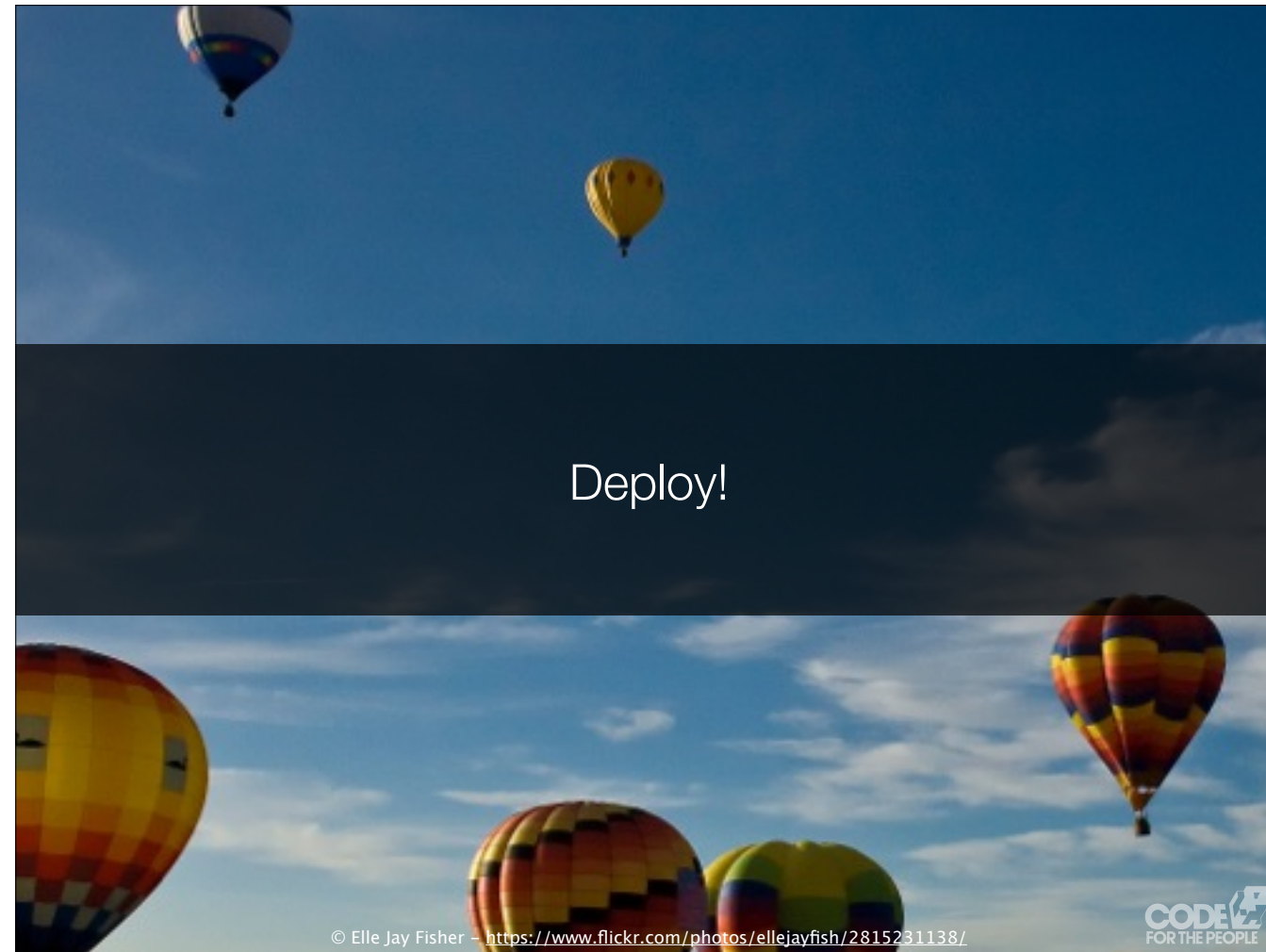
Composer has given us good flexibility in allowing us to have all our nicely separated repositories, but now we need to bring everything together; perhaps into a single repository so we can use Git push (like with WPEngine) , perhaps to SFTP up.



Enter our build script. The build script pulls together all of our requirements, pulls down a Git repository which we use to keep track of all deployed changes...



...and packages it up neatly to go. We can now SFTP, or Git push to deploy, as the hosting company allows.



We have lift off.



So, are we there yet? The system is solid, the way we work with it needs to be finessed.

- * We need to refined, solid, tested processes
- * Most importantly, we need solid documentation, to ensure people know how to work through the processes



P.S. we're looking for developers
(simonw@codeforthepeople.com)

© Yaniv Golan - <https://www.flickr.com/photos/yanivg/238454698/>

CODE
FOR THE PEOPLE



Thanks for listening (or reading)!