Perl and Docker, sitting in a tree

Jon Allen (JJ) – jj@opusvl.com

OPUS VL

Docker

Packages an application and dependencies into a portable container

Simplifies workflow

Use same containers for development and production

WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

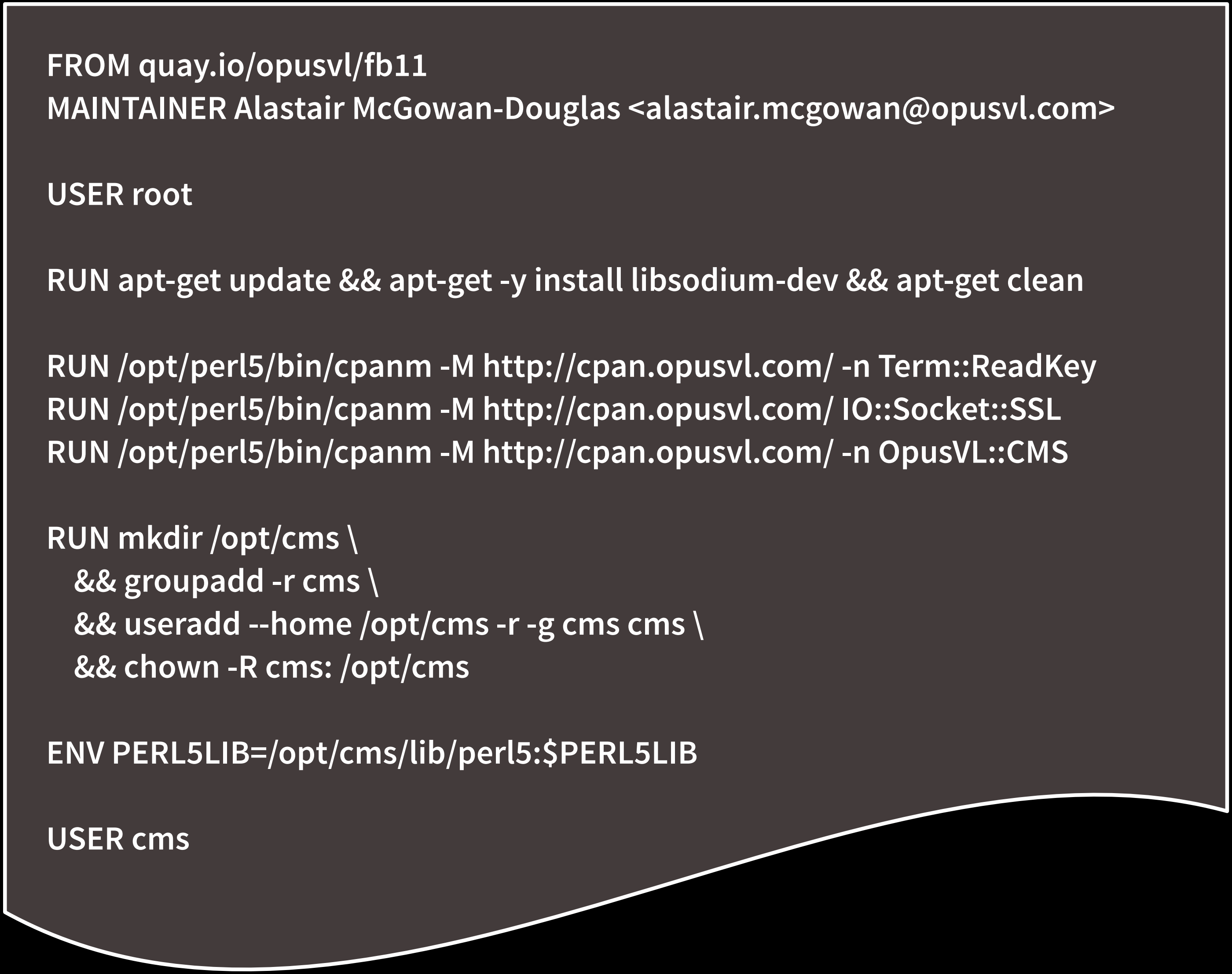
“Works on my machine”

“So of course it will work in production”

Concepts

Dockerfile

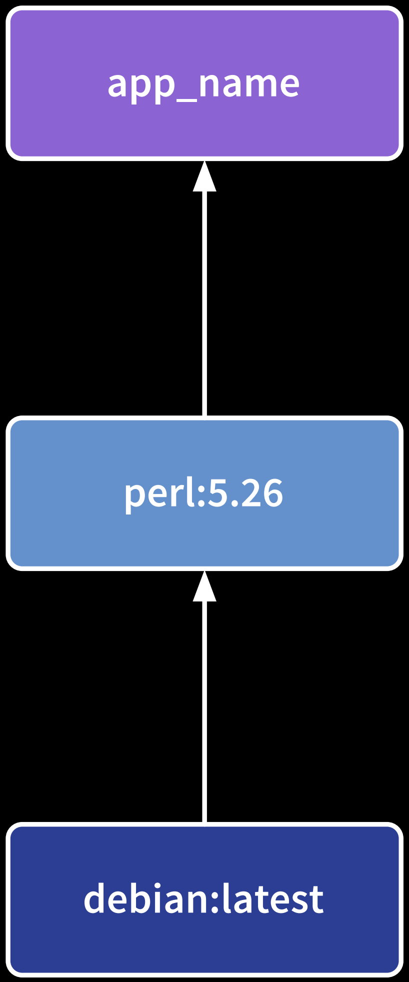
Set of instructions to build an image



```
FROM quay.io/opusvl/fb11
MAINTAINER Alastair McGowan-Douglas <alastair.mcgowan@opusvl.com>

USER root

RUN apt-get update && apt-get -y install libsodium-dev && apt-get clean

RUN /opt/perl5/bin/cpanm -M http://cpan.opusvl.com/ -n Term::ReadKey
RUN /opt/perl5/bin/cpanm -M http://cpan.opusvl.com/ IO::Socket::SSL
RUN /opt/perl5/bin/cpanm -M http://cpan.opusvl.com/ -n OpusVL::CMS

RUN mkdir /opt/cms \
   && groupadd -r cms \
   && useradd --home /opt/cms -r -g cms cms \
   && chown -R cms: /opt/cms

ENV PERL5LIB=/opt/cms/lib/perl5:$PERL5LIB

USER cms
```

Layers

Inheritance and image re-use

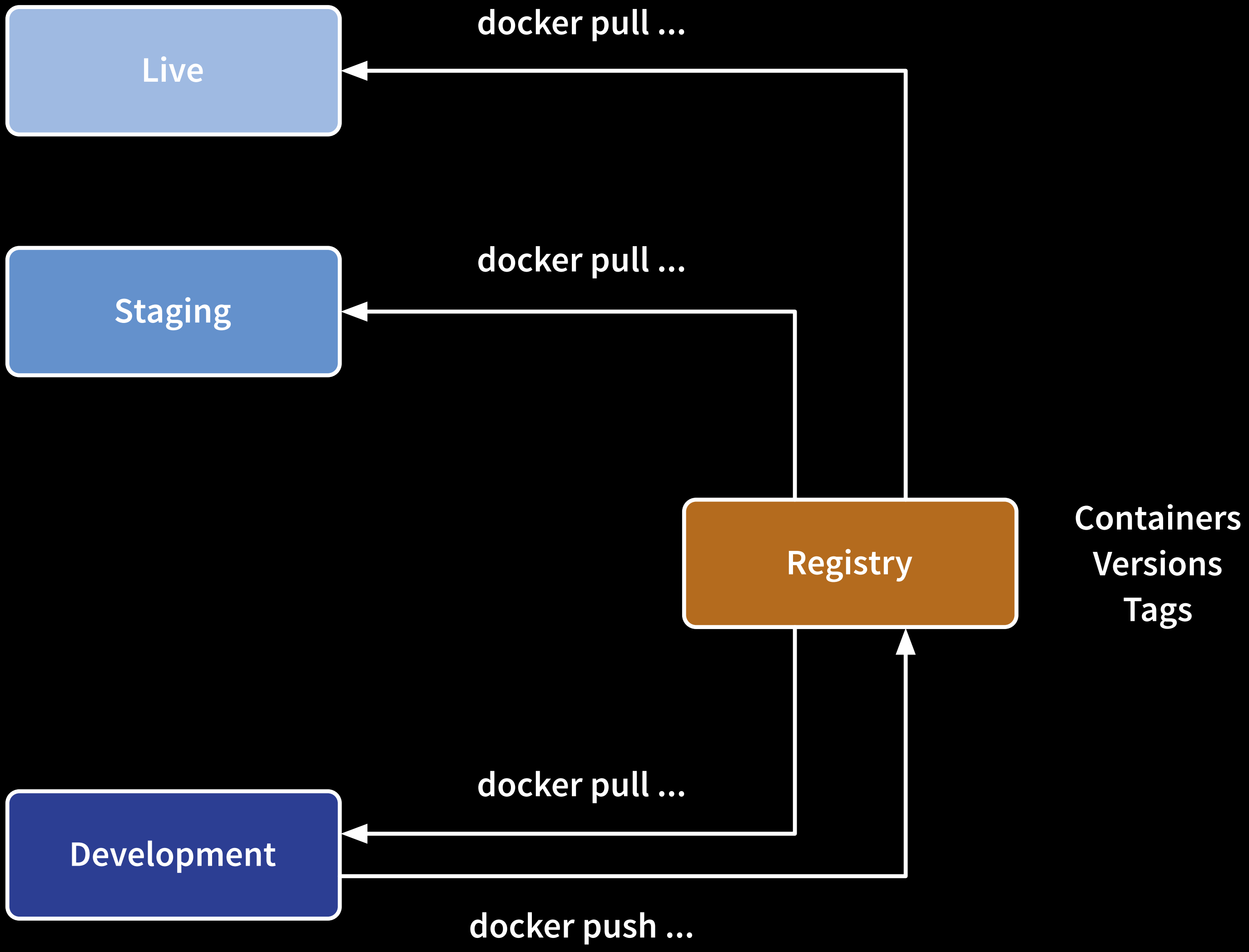


```
FROM perl:5.26
...
# install application
```

```
FROM debian:latest
...
# install Perl
```

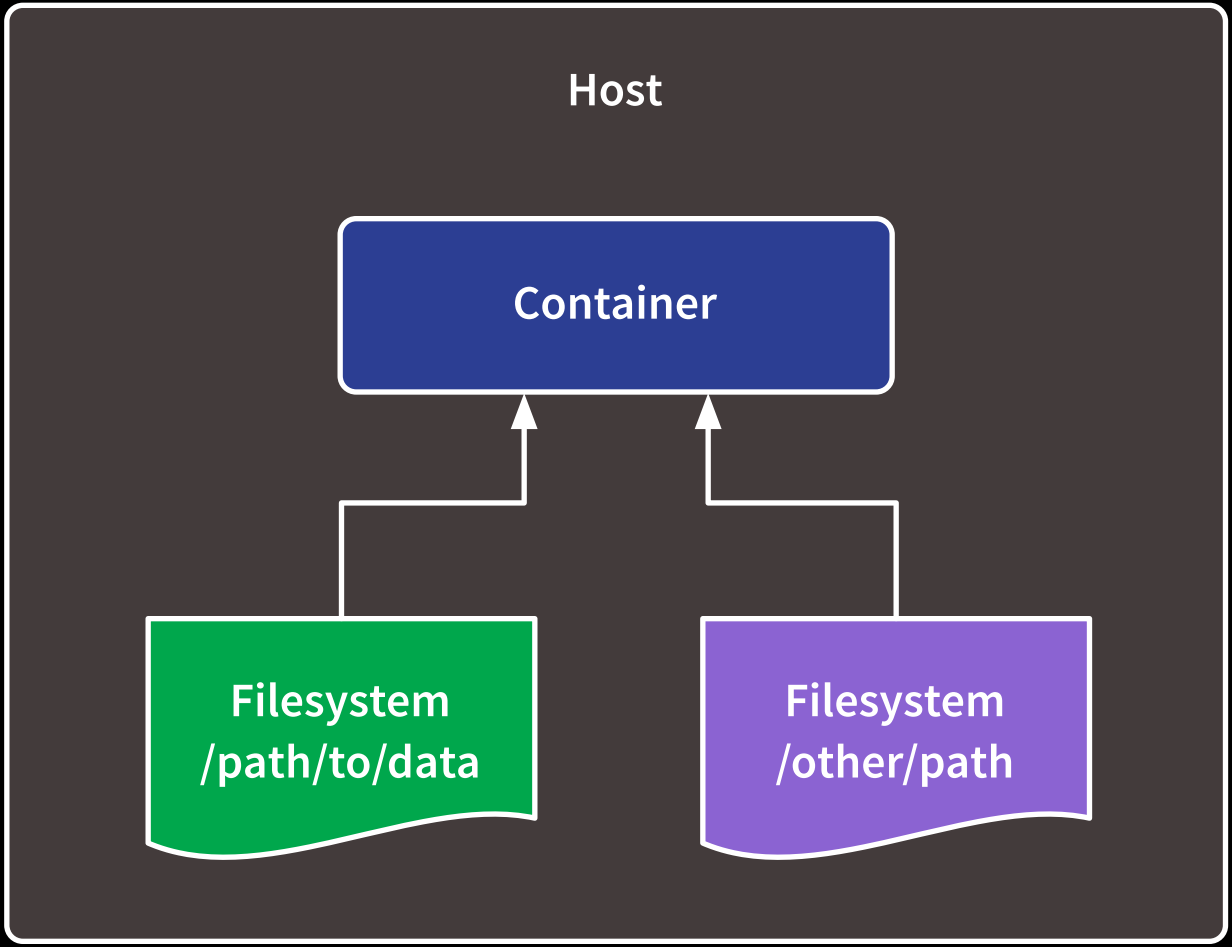
Registry

Version controlled repository of layers and images


docker pull ...
Live
docker pull ...
Staging
Registry
Containers
Versions
Tags
docker pull ...
Development
docker push ...

Volumes

Store persistent data outside of the container


Host
Container
Filesystem
/path/to/data
Filesystem
/other/path

Entrypoint

Command that is run when the container starts

Docker and Perl

Build a container from an application

Application runs when the container starts

Perl module distribution with a .psgi file

Dockerfile included in the application source tree

```
MyApp
    /Makefile.PL
    /README
    /lib
    /bin/myapp.psgi
    /Dockerfile.base
    /Dockerfile.patch
    /vendor
```

Dockerfile.base

Builds from a standard OS image

Installs the application and all its dependencies

Time passes…

Half of CPAN downloads

Dockerfile.patch

Builds on top of Base image

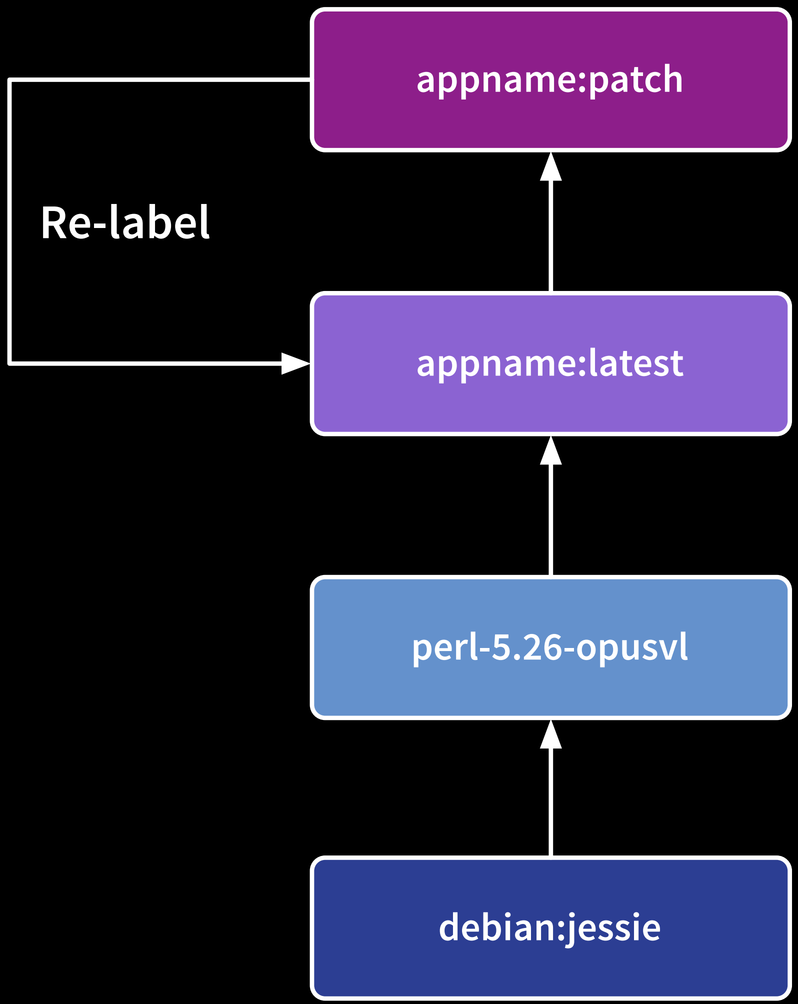
Installs application .tar.gz from disk

Can build a test container without releasing to CPAN

Installs contents of /vendor directory

Allows dependency modules to be patched

Layers



Patch layer for maintenance / test releases between full rebuilds of Base (add tarball to vendor directory)

Base layer = full build of the application (installs all dependencies using cpanm)

Perl with standard tools (e.g. build-essential, development libraries etc)

Pin base OS to specific version (jessie)

Entrypoint

Runs the application .psgi file

OpusVL::Docker

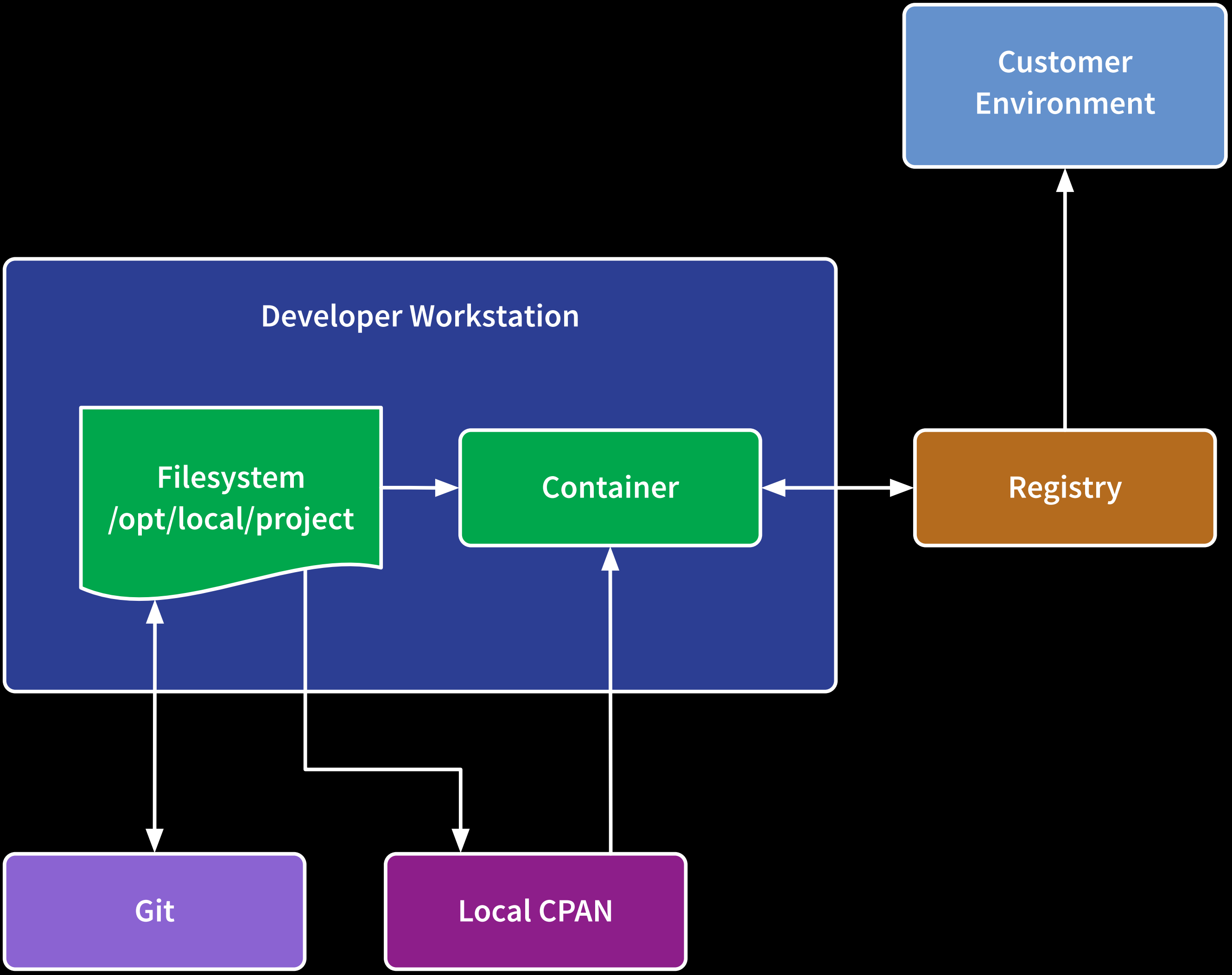
Add to Makefile.PL as a dependency

Includes entrypoint script and template Dockerfiles

Developer workflow

Use same containers for development and production

Mount local Git repos and inject into container, overriding the installed code


Customer Environment
Developer Workstation
Filesystem /opt/local/project
Container
Registry
Git
Local CPAN

entrypoint

If $ENV{DEV_MODE}
is set…

Searches for volumes

/opt/local/project/dist/lib
/opt/local/*/*/lib

Adds each lib/ directory found to $ENV{PERL5LIB}

Check out all repos to /opt/local/projectname/*

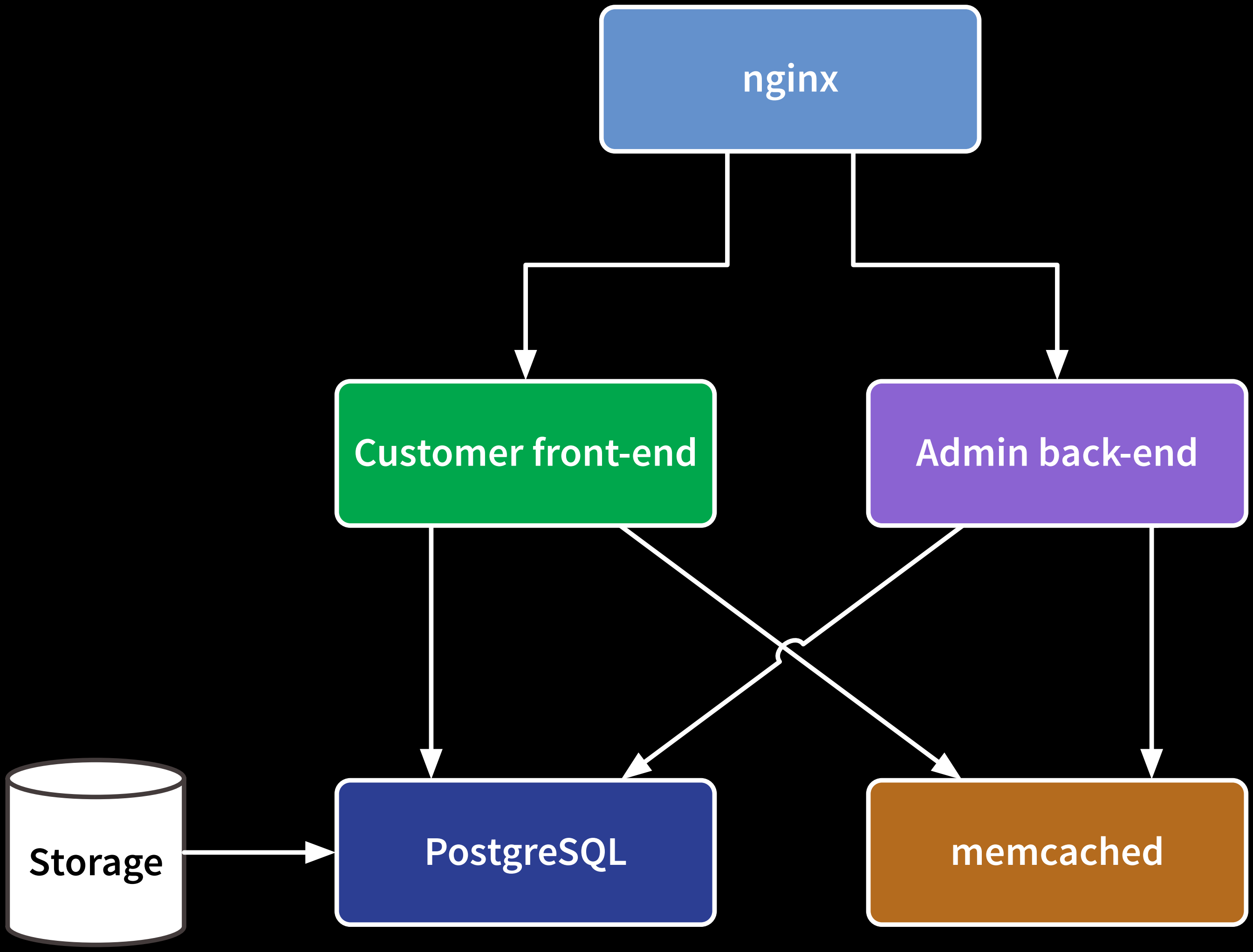
Mount volumes in docker-compose.override.yml

Automatically installs dependencies if $ENV{INSTALLDEPS} set

Uses plackup instead of Starman / Martian (single worker process)

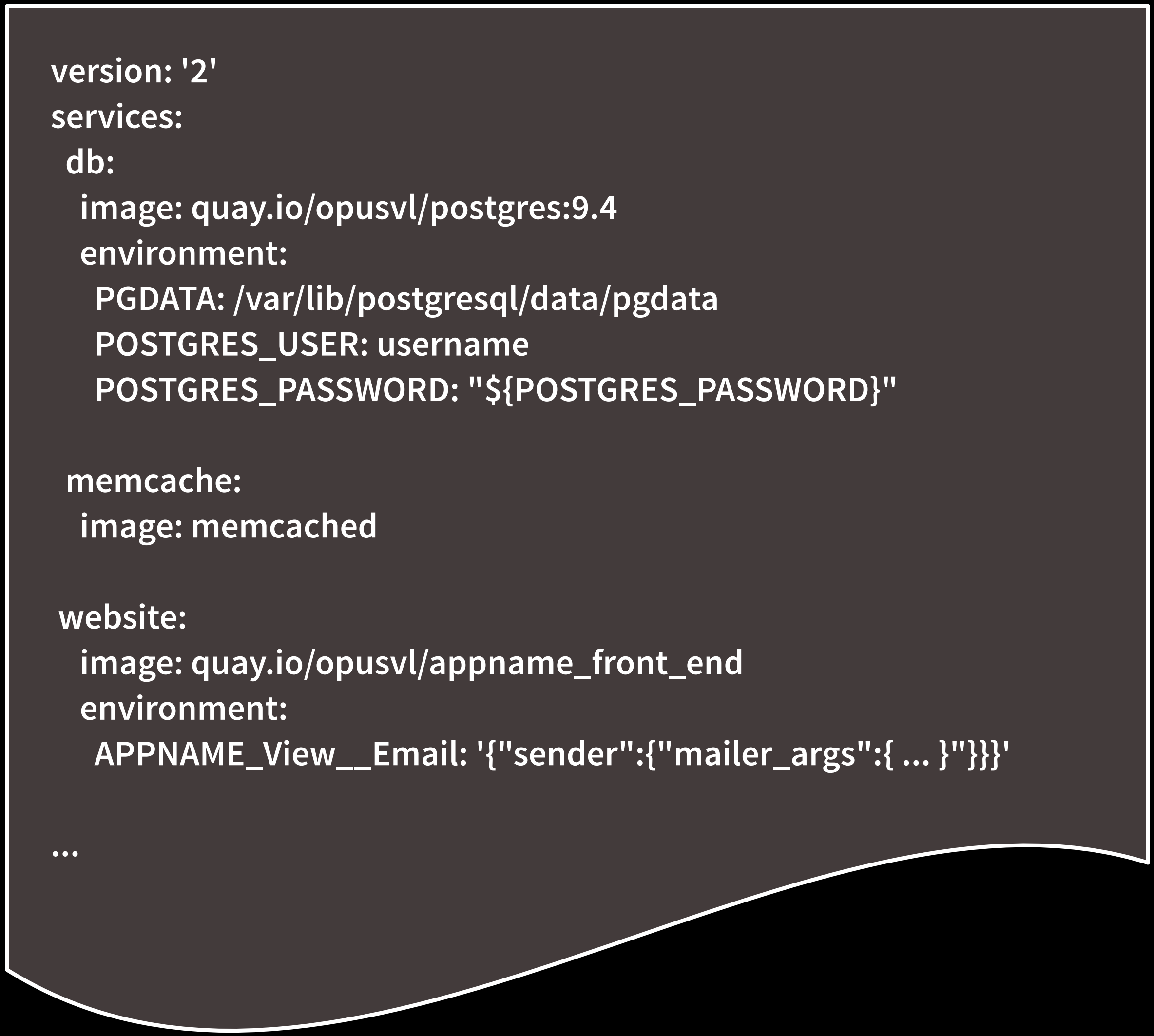
Docker Compose

Because systems have moving parts


nginx
Customer front-end
Admin back-end
Storage
PostgreSQL
memcached

Compose

Wiring and configuration of environment

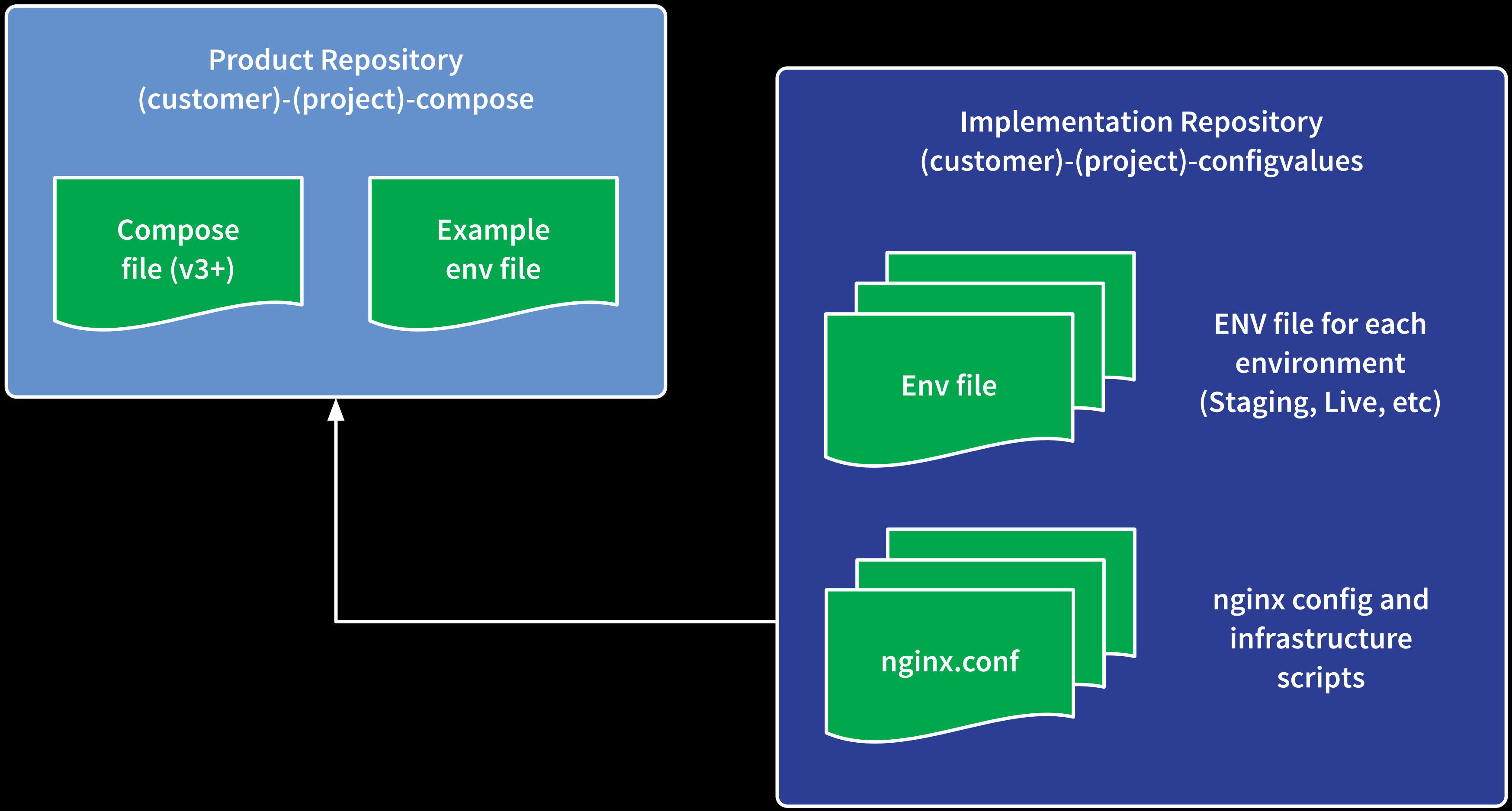


```
version: '2'
services:
 db:
  image: quay.io/opusvl/postgres:9.4
  environment:
   PGDATA: /var/lib/postgresql/data/pgdata
   POSTGRES_USER: username
   POSTGRES_PASSWORD: "${POSTGRES_PASSWORD}"

 memcache:
  image: memcached

 website:
  image: quay.io/opusvl/appname_front_end
  environment:
   APPNAME_View__Email: '{"sender":{"mailer_args":{ ... }"}}}'

...
```

Not part of the application Perl module

Git repositories;

- product
- implementation



Product Repository
(customer)-(project)-compose

Compose
file (v3+)

Example
env file

Implementation Repository
(customer)-(project)-configvalues

Env file

ENV file for each
environment
(Staging, Live, etc)

nginx.conf

nginx config and
infrastructure
scripts

Same compose file everywhere

(dev, staging, live)

Configuration in source control

Sensitive data protected with Docker Secrets

OpusVL::Docker

github.com/
OpusVL/Perl-Docker