goto; chicago

Bash and All That

Dean Wampler
@deanwampler

follow us @gotochgo

goto; chicago

Click ‘Rate Session’ to rate session and ask questions.

follow us @gotochgo

goto; chicago

Please

Remember to rate this session

Thank you!

follow us @gotochgo

goto;
chicago

Click ‘Rate Session’

Rate **5** sessions to get the supercool GOTO reward

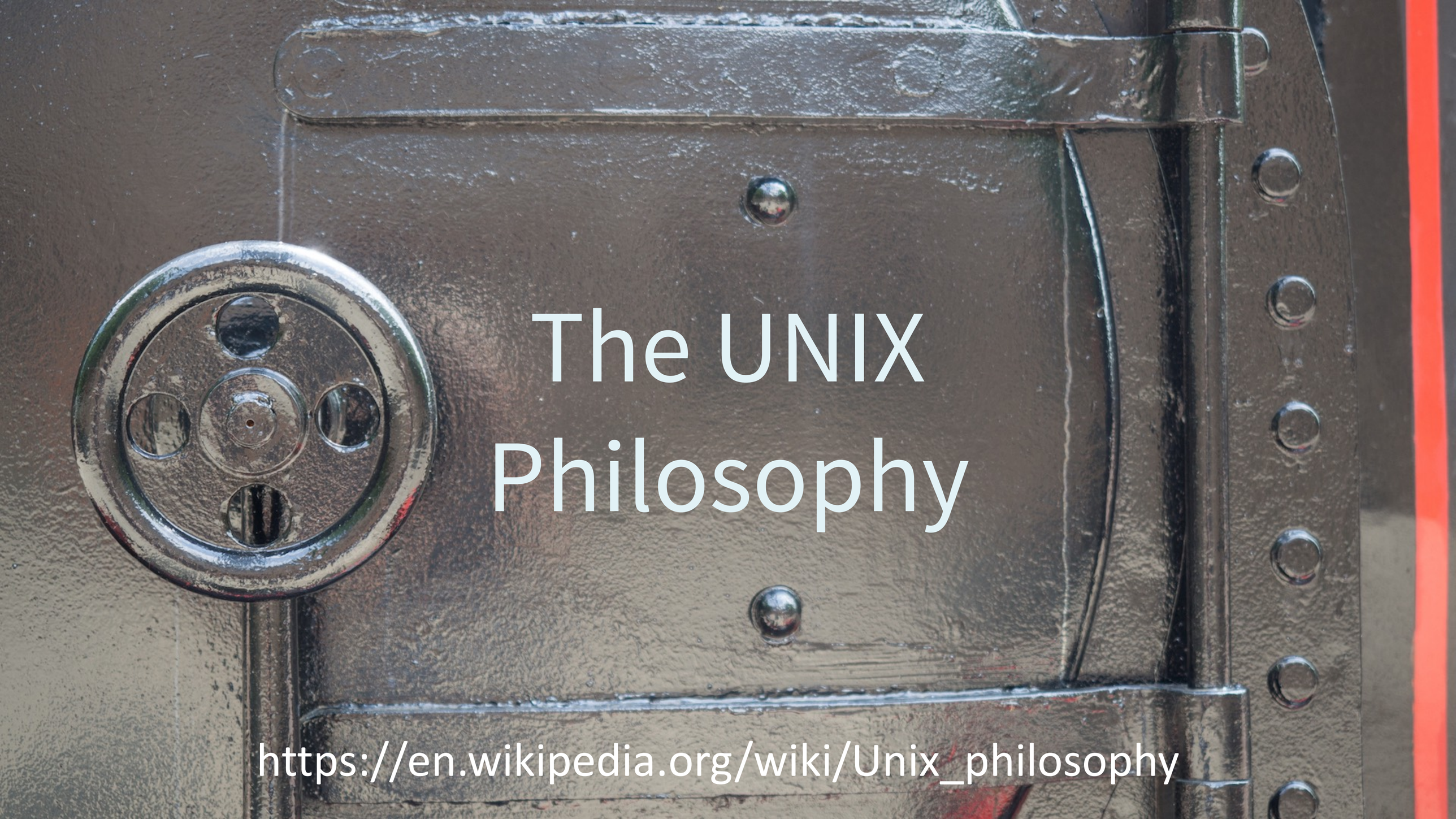
follow us @gotochgo

Bash and All That

Why Ancient *NIX Tools Are Still Essential

Dean Wampler, Ph.D.
dean@lightbend.com
@deanwampler

Lightbend



The UNIX Philosophy

https://en.wikipedia.org/wiki/Unix_philosophy

- Make each program do one thing well

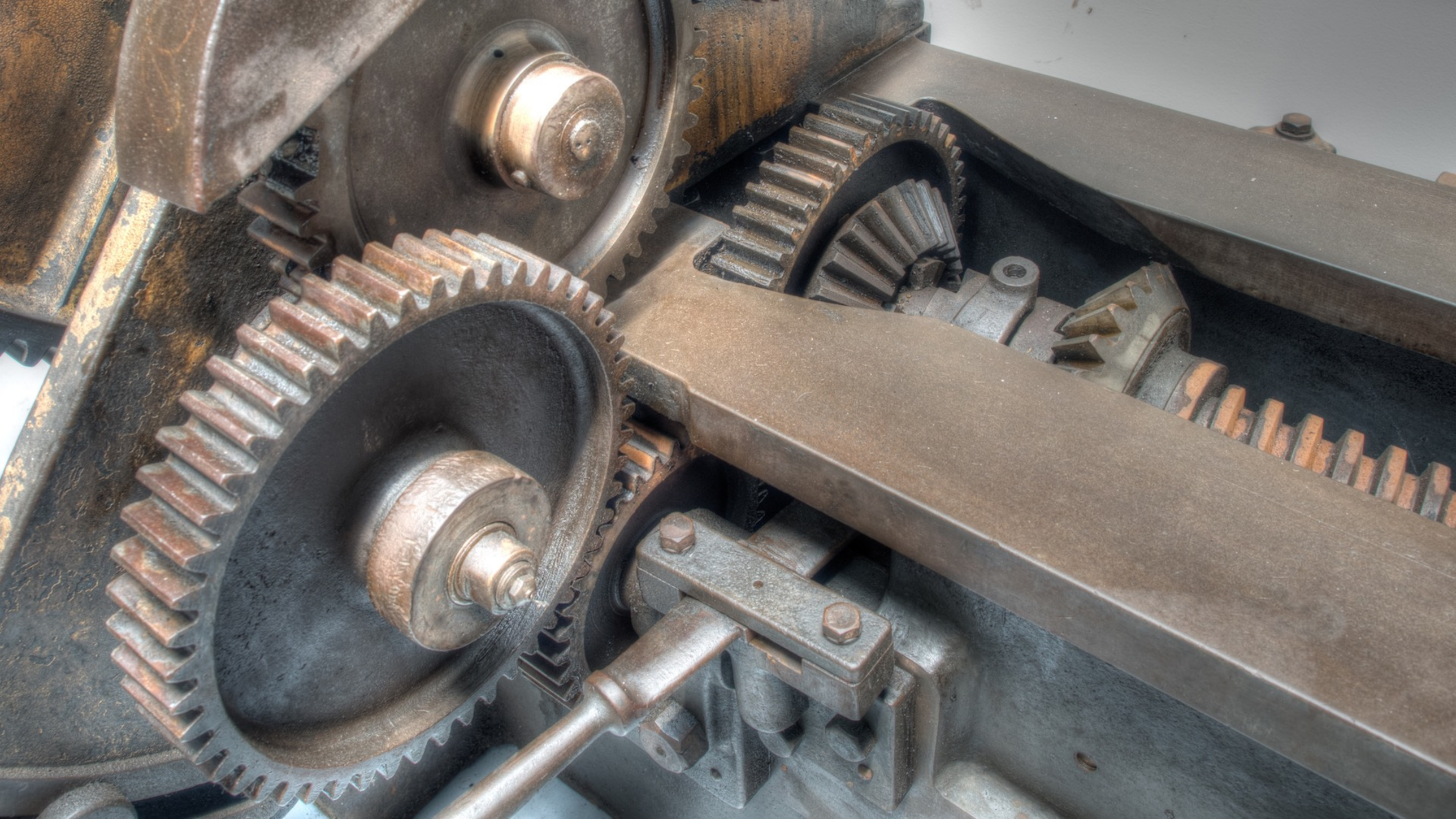
- Expect the output of every program to become the input to another program

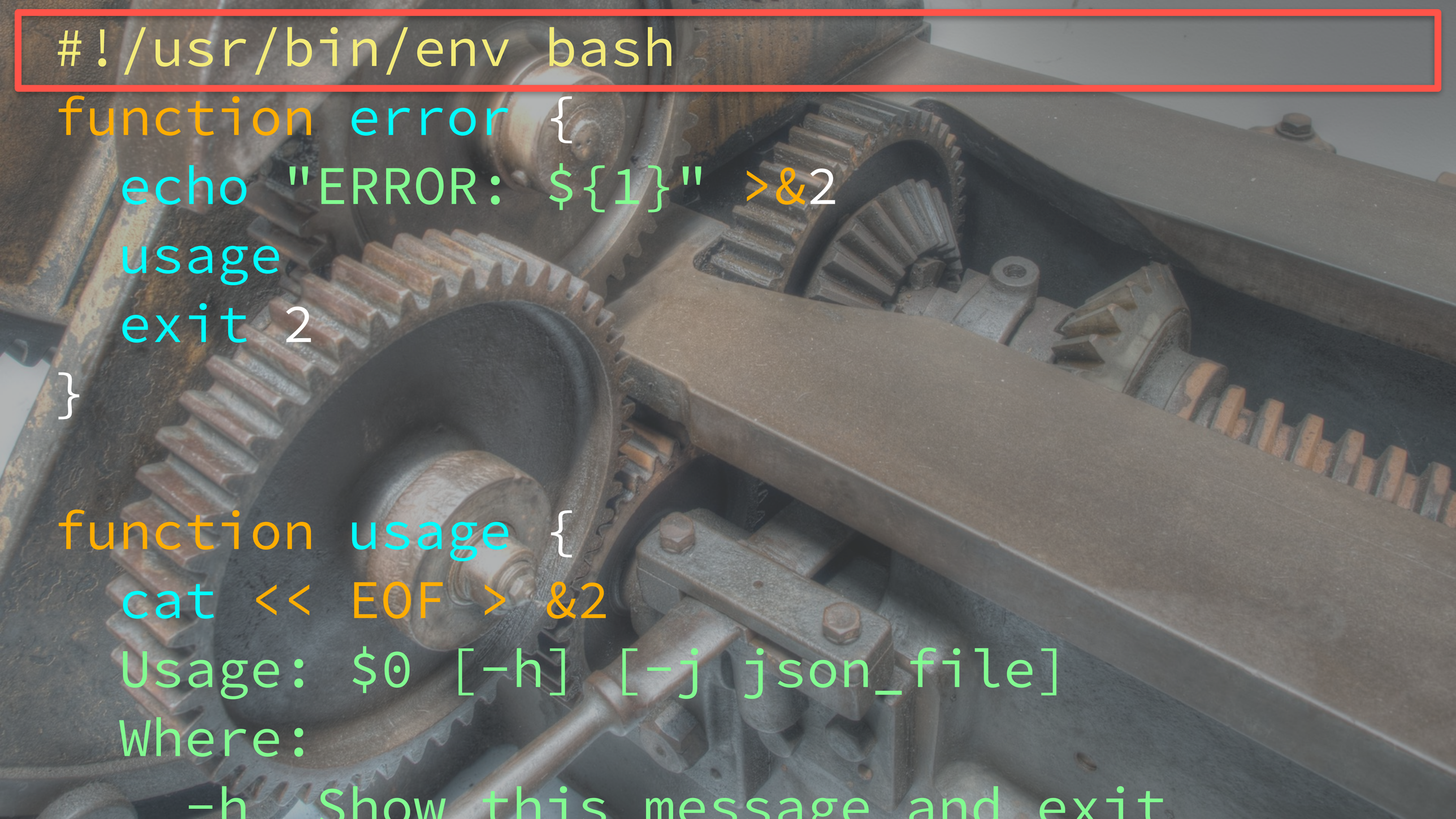
- Design and build software to be tried early, ideally within weeks

- Use tools in preference to unskilled help to lighten a programming task

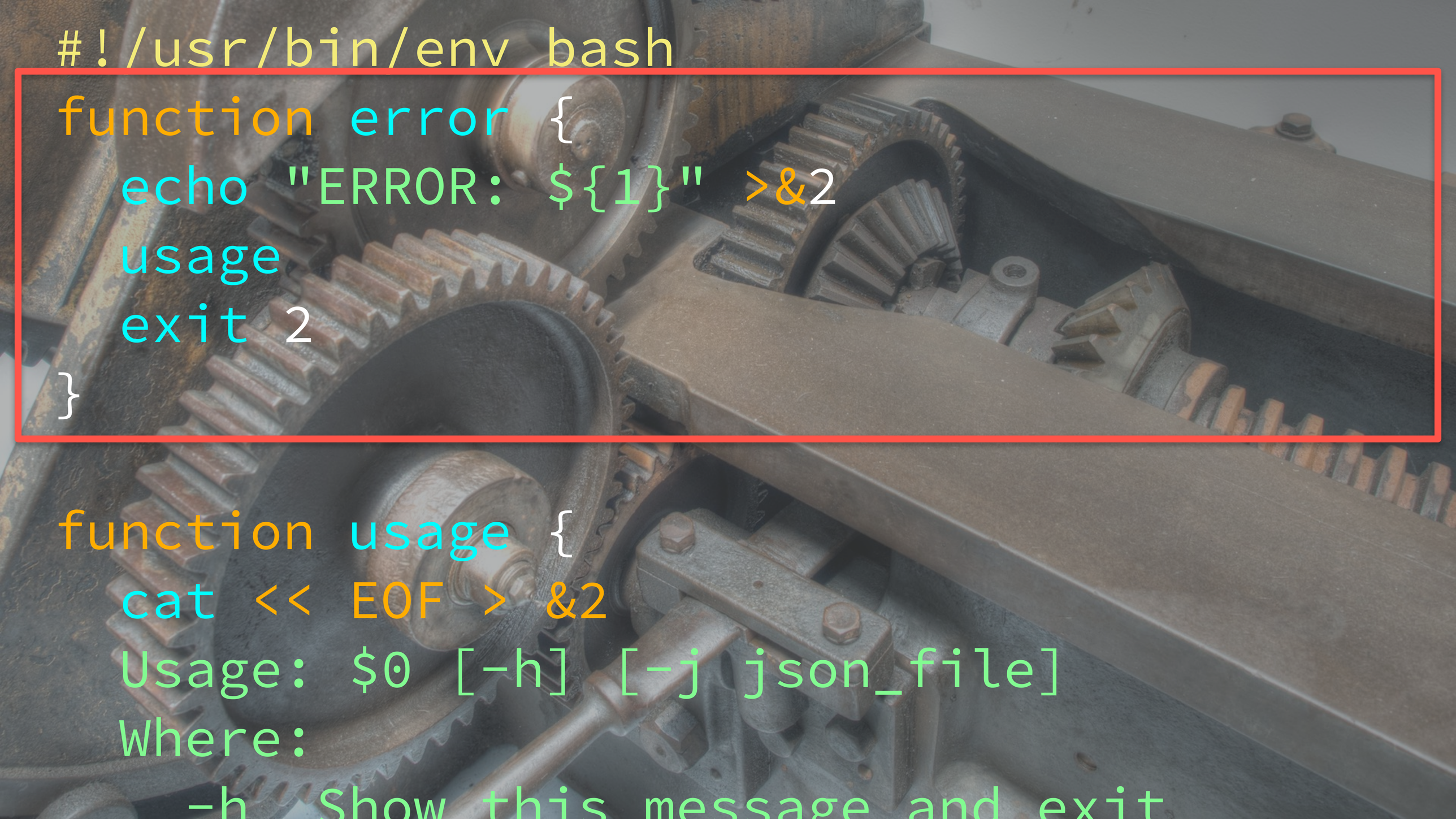

tools...


bash
for scripting

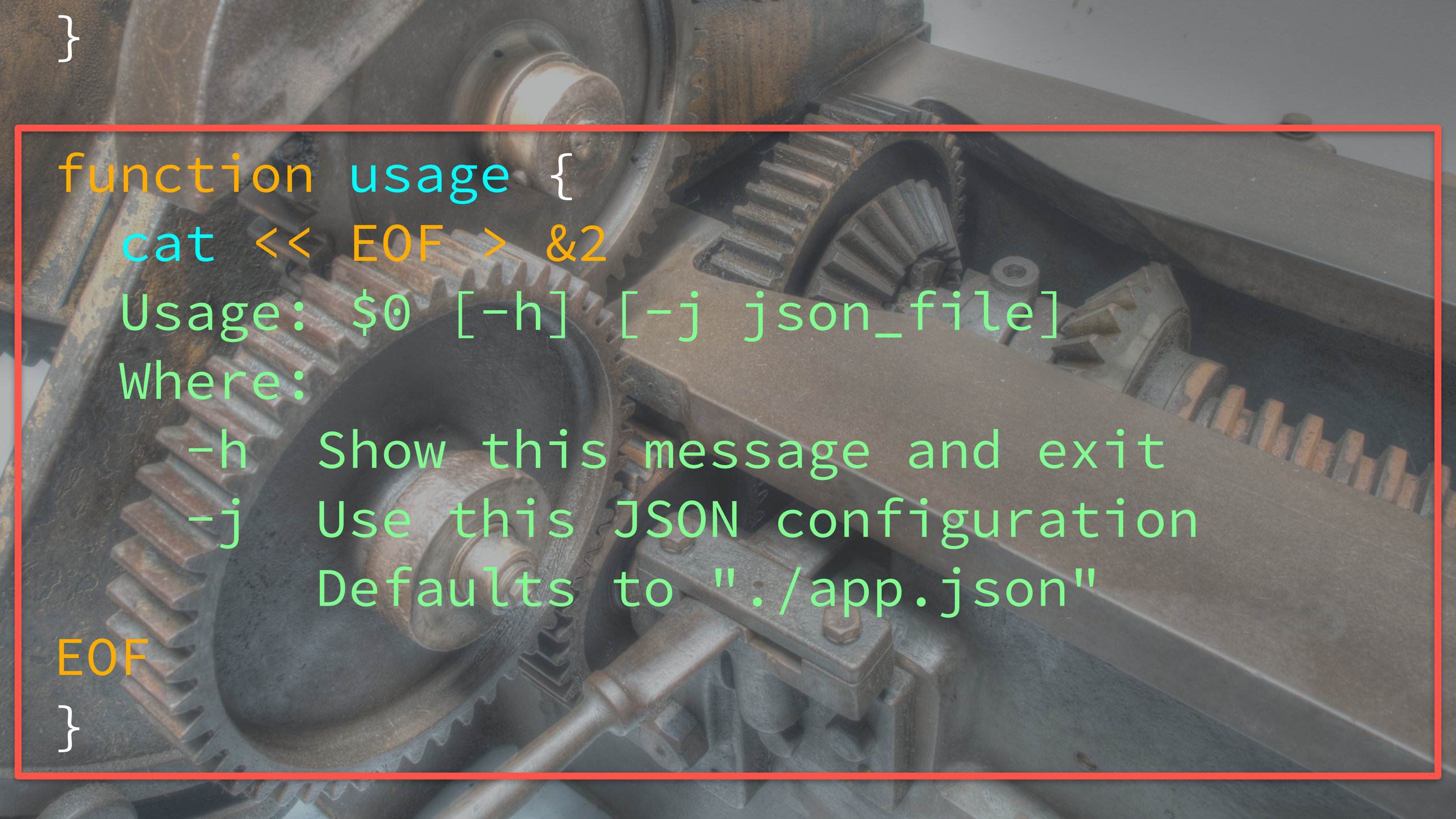




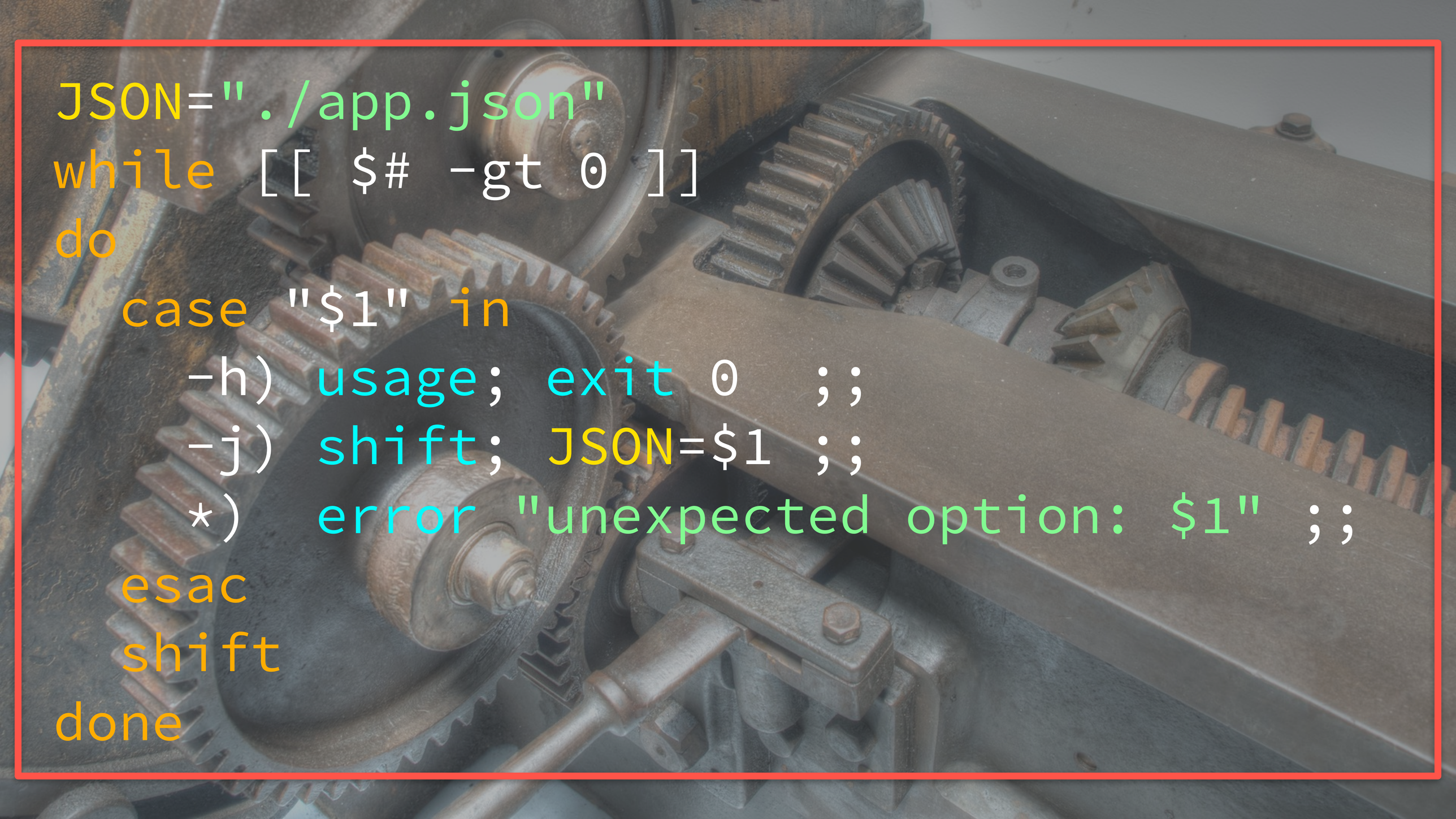
```
#!/usr/bin/env bash
function error {
  echo "ERROR: ${1}" >&2
  usage
  exit 2
}

function usage {
  cat << EOF > &2
  Usage: $0 [-h] [-j json_file]
  Where:
    -h  Show this message and exit
```



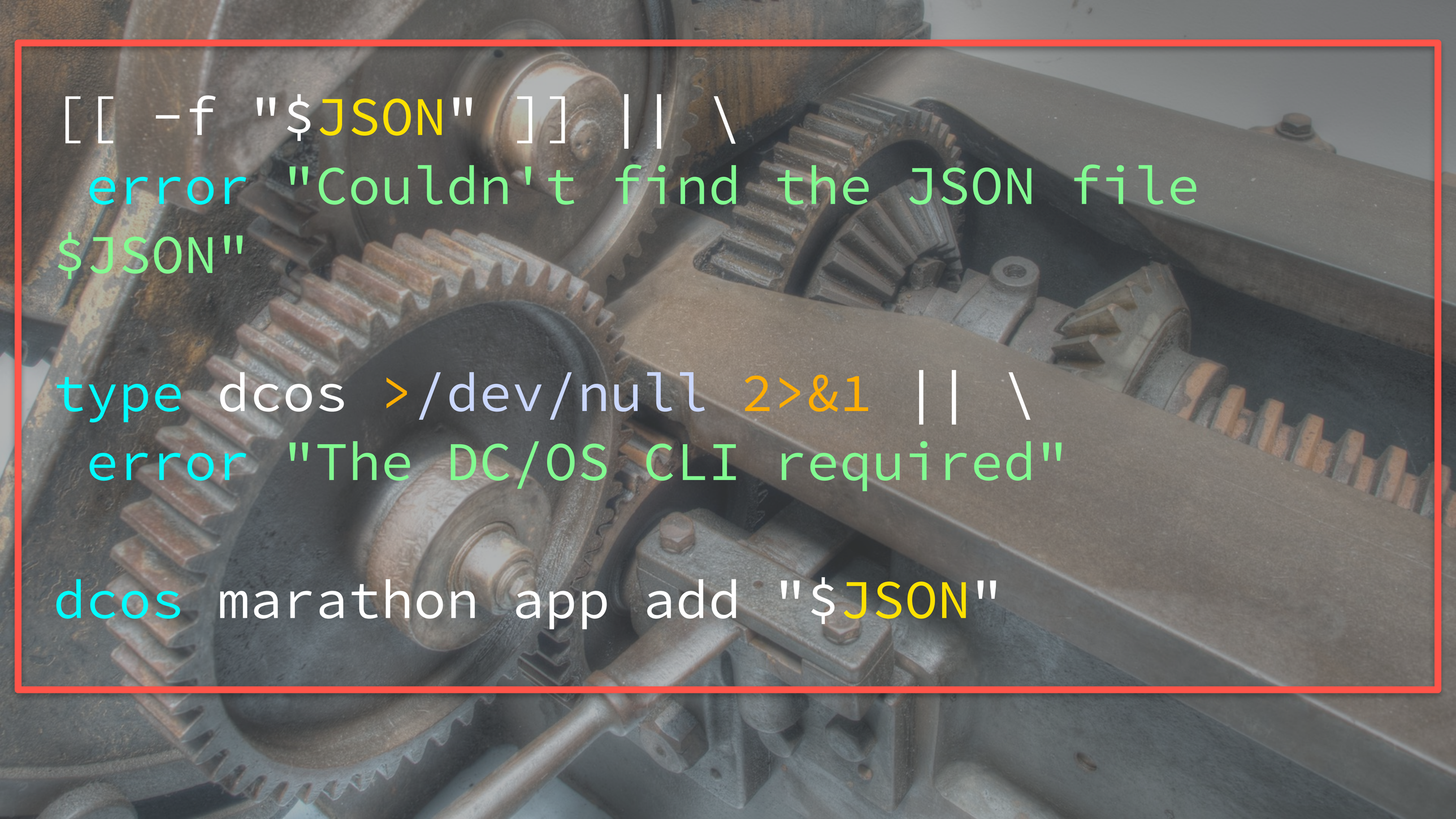
```
#!/usr/bin/env bash
function error {
  echo "ERROR: ${1}" >&2
  usage
  exit 2
}

function usage {
  cat << EOF > &2
  Usage: $0 [-h] [-j json_file]
  Where:
    -h  Show this message and exit
```



```
}

function usage {
  cat << EOF > &2
  Usage: $0 [-h] [-j json_file]
  Where:
    -h   Show this message and exit
    -j   Use this JSON configuration
         Defaults to "./app.json"
EOF
}
```



```
JSON="./app.json"
while [[ $# -gt 0 ]]
do
  case "$1" in
    -h) usage; exit 0  ;;
    -j) shift; JSON=$1 ;;
    *)  error "unexpected option: $1" ;;
  esac
  shift
done
```

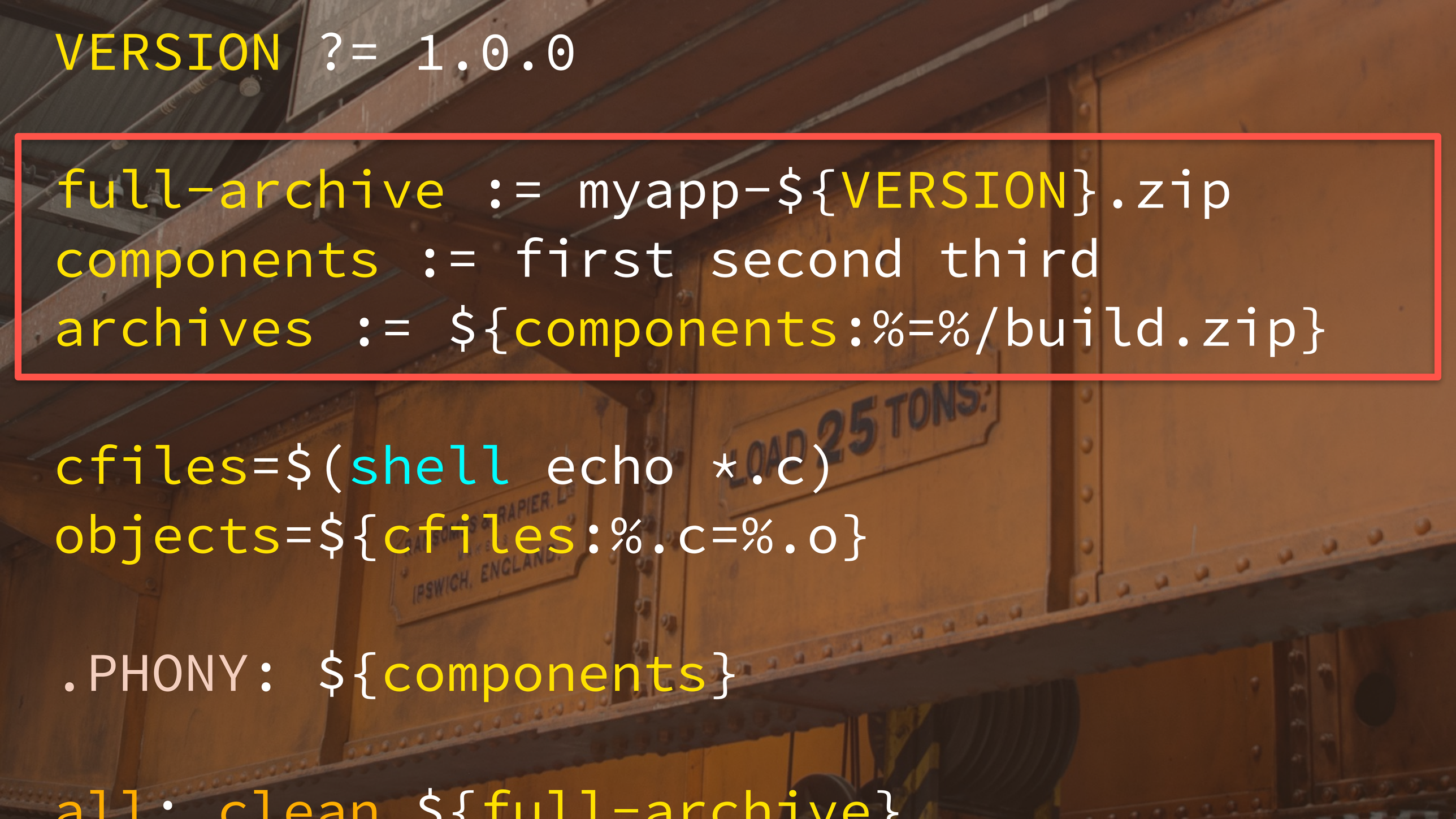

[[-f "$JSON"]] || \
error "Couldn't find the JSON file
$JSON"
type dcos >/dev/null 2>&1 || \
error "The DC/OS CLI required"
dcos marathon app add "$JSON"

- Low ceremony
  - Pros and cons of "stringly typed"
- Composability
- Productivity

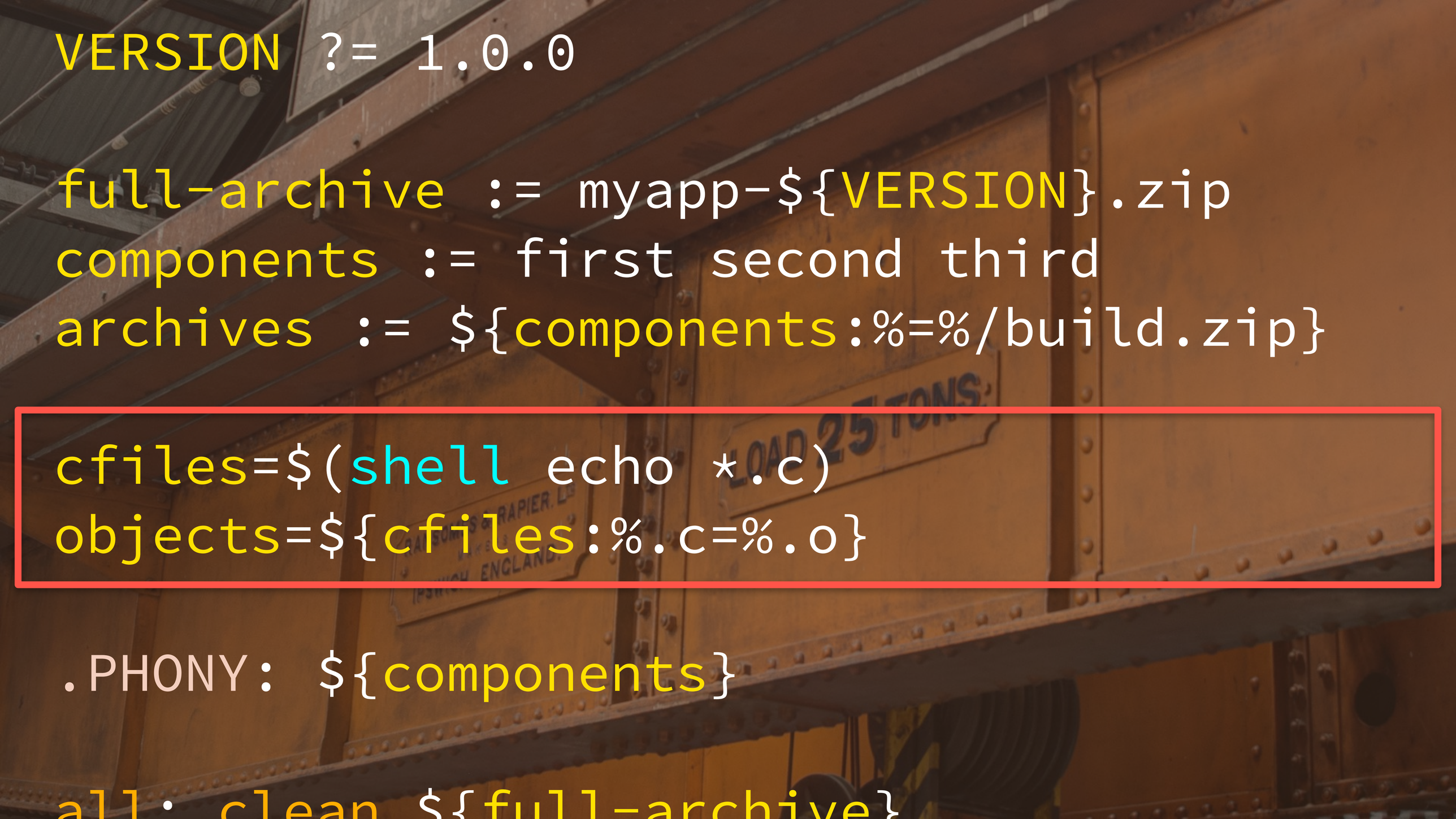

make
for workflows


AUX
RANSOMES & RAPIER, LTD
MAKERS
IPSWICH, ENGLAND.
LOAD 25 TONS.

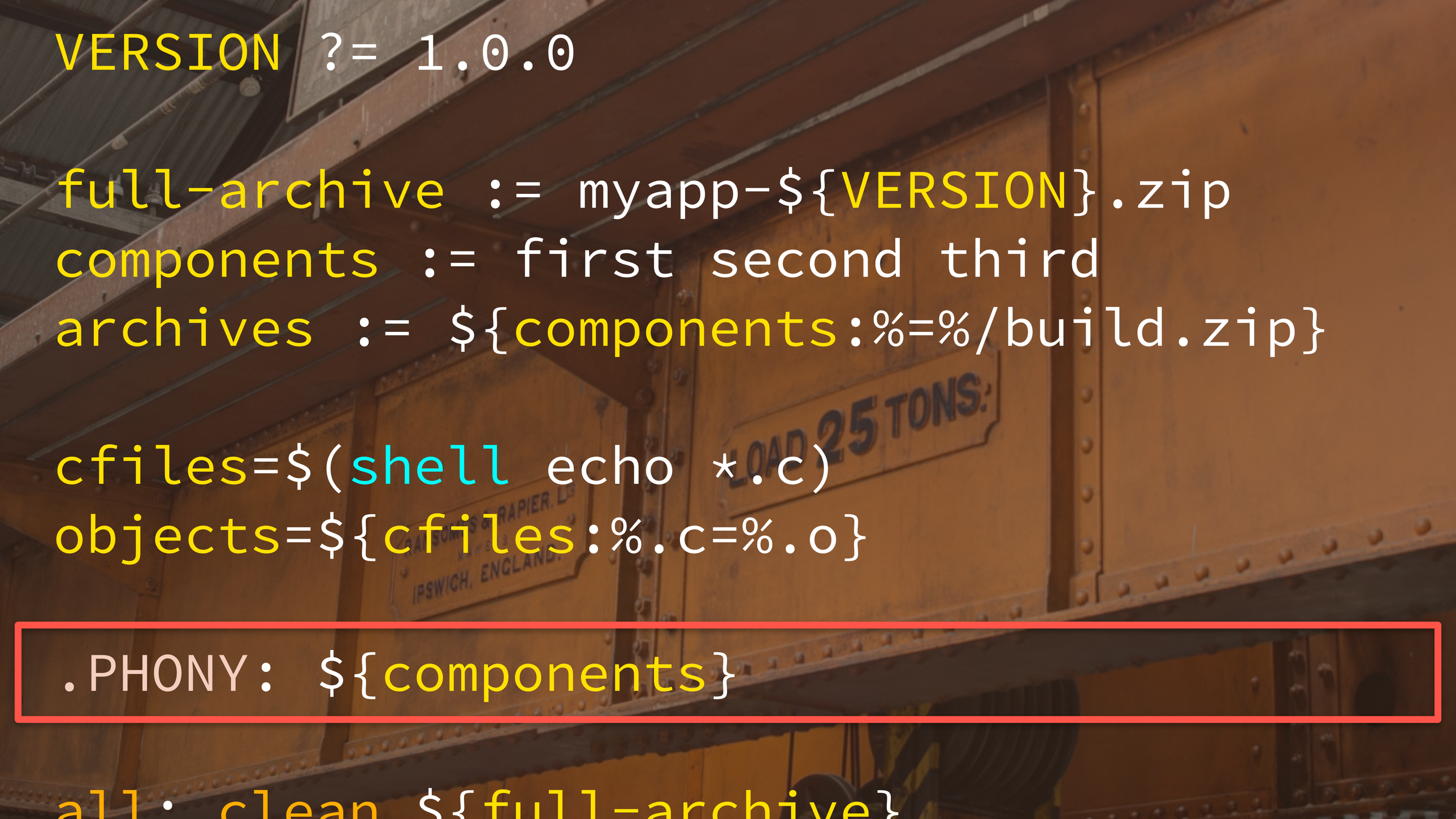
```
VERSION ?= 1.0.0

full-archive := myapp-${VERSION}.zip
components := first second third
archives := ${components:%=%/build.zip}

cfiles=$(shell echo *.c)
objects=${cfiles:%.c=%.o}

.PHONY: ${components}

all: clean ${full-archive}
```



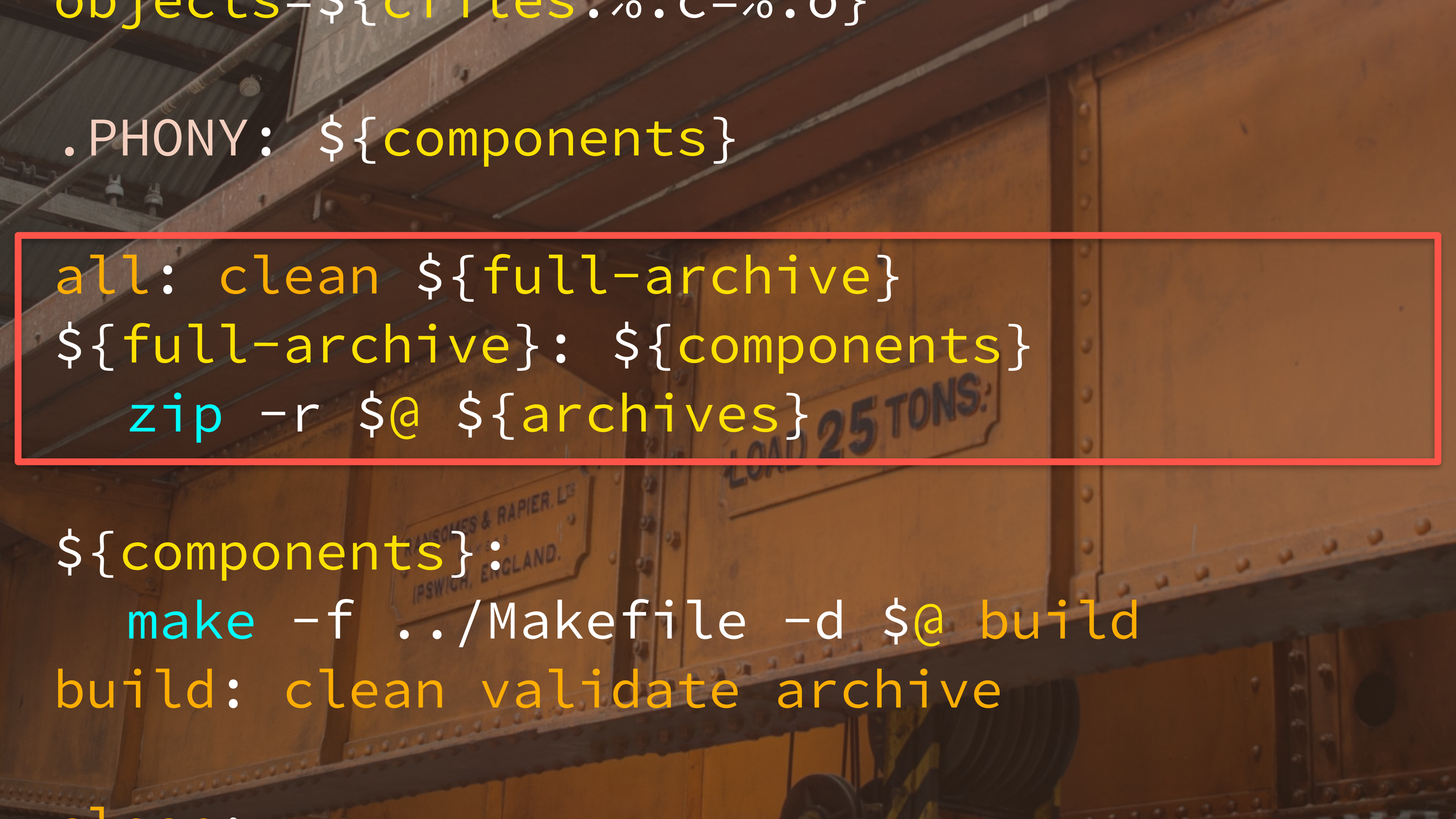
```
VERSION ?= 1.0.0

full-archive := myapp-${VERSION}.zip
components := first second third
archives := ${components:%=%/build.zip}

cfiles=$(shell echo *.c)
objects=${cfiles:%.c=%.o}

.PHONY: ${components}

all: clean ${full-archive}
```

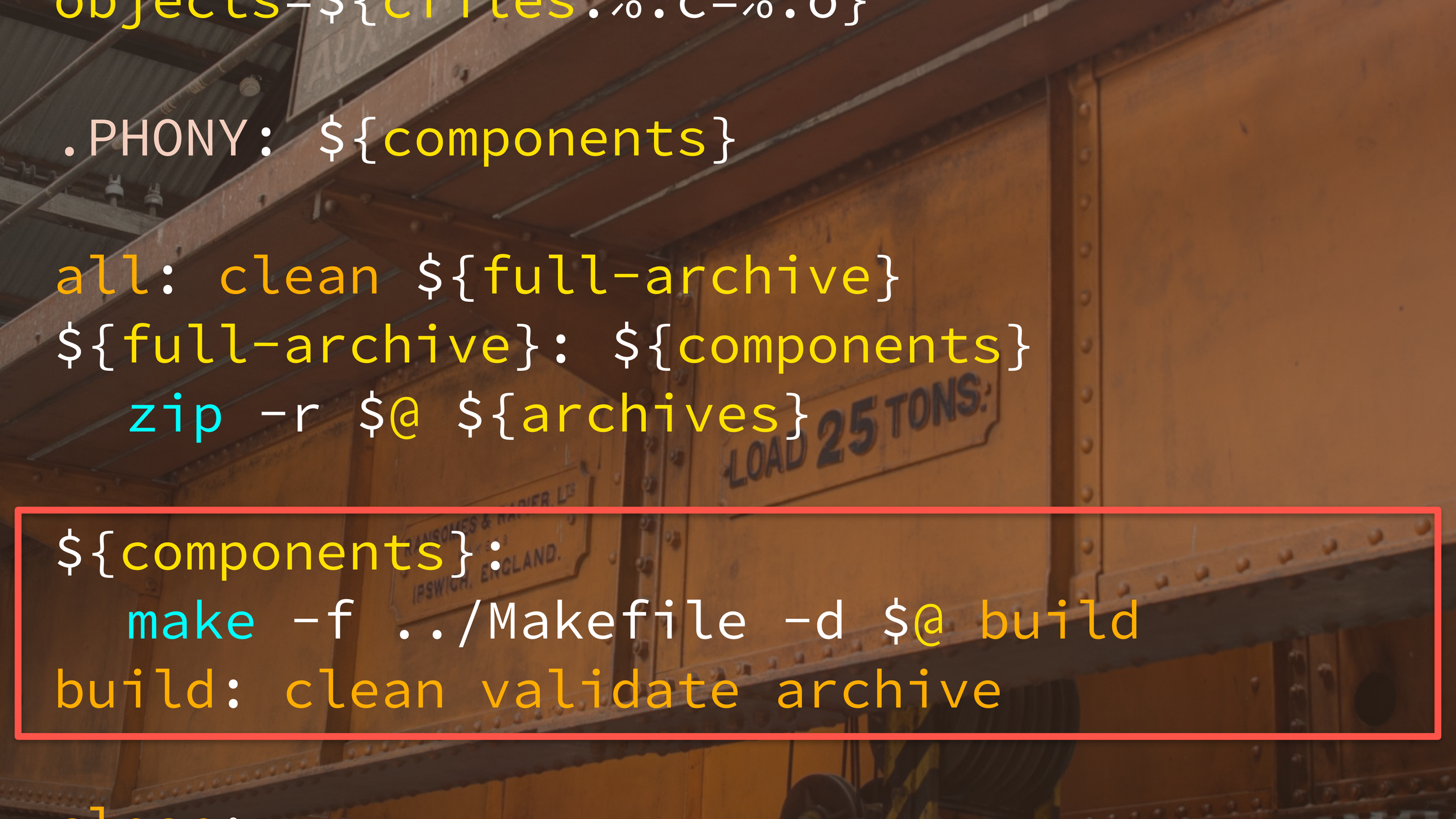


```
VERSION ?= 1.0.0

full-archive := myapp-${VERSION}.zip
components := first second third
archives := ${components:%=%/build.zip}

cfiles=$(shell echo *.c)
objects=${cfiles:%.c=%.o}

.PHONY: ${components}

all: clean ${full-archive}
```

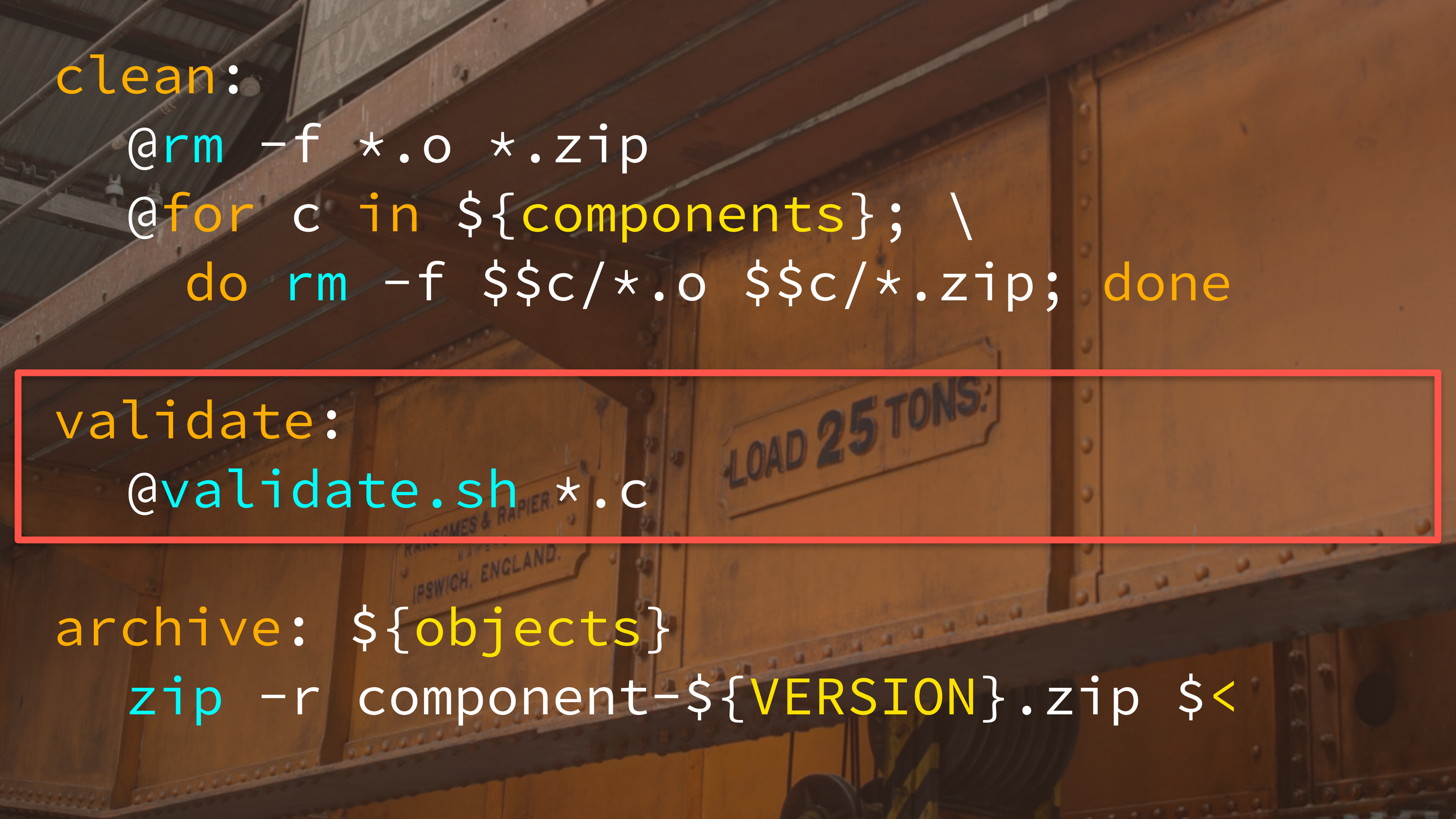


```
VERSION ?= 1.0.0

full-archive := myapp-${VERSION}.zip
components := first second third
archives := ${components:%=%/build.zip}

cfiles=$(shell echo *.c)
objects=${cfiles:%.c=%.o}

.PHONY: ${components}

all: clean ${full-archive}
```

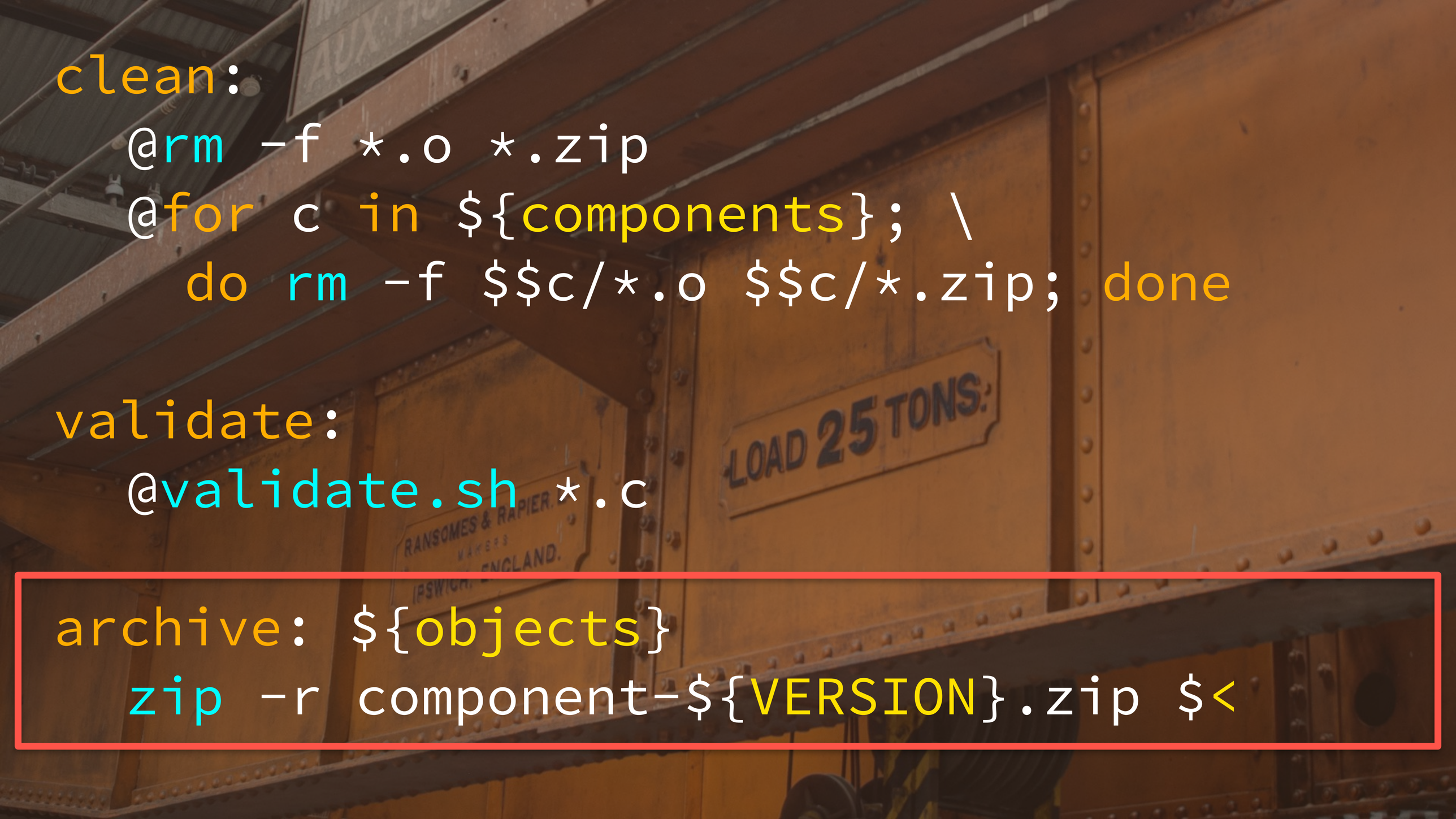

objects=${cfiles:%.c=%.o}
.PHONY: ${components}
all: clean ${full-archive}
${full-archive}: ${components}
zip -r $@ ${archives}
${components}:
make -f ../Makefile -d $@ build
build: clean validate archive


objects=${cfiles:%.c=%.o}
.PHONY: ${components}
all: clean ${full-archive}
${full-archive}: ${components}
zip -r $@ ${archives}
${components}:
make -f ../Makefile -d $@ build
build: clean validate archive


```
clean:
	@rm -f *.o *.zip
	@for c in ${components}; \
	  do rm -f $$c/*.o $$c/*.zip; done

validate:
	@validate.sh *.c

archive: ${objects}
	zip -r component-${VERSION}.zip $<
```

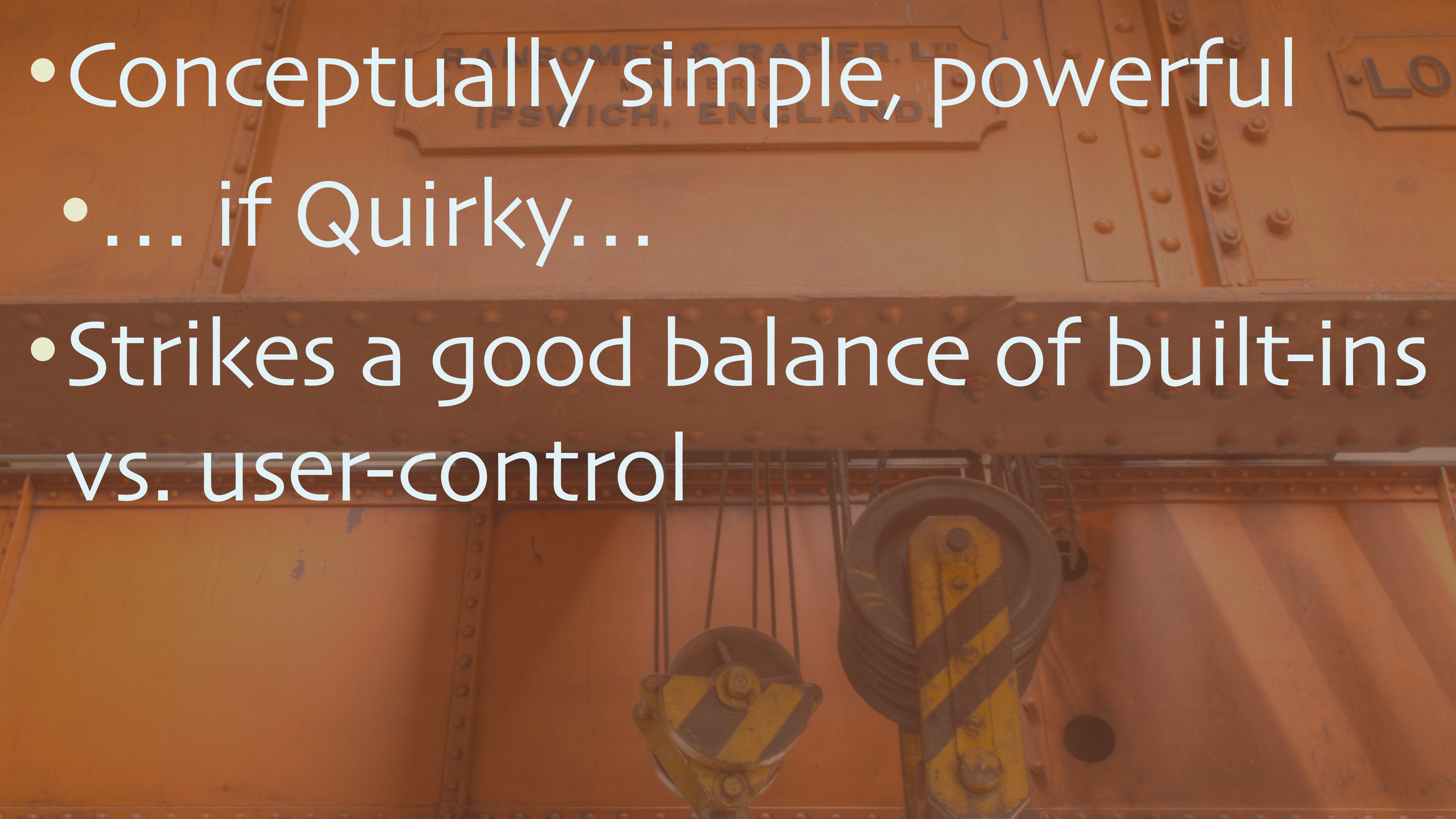



```
clean:
	@rm -f *.o *.zip
	@for c in ${components}; \
	  do rm -f $$c/*.o $$c/*.zip; done

validate:
	@validate.sh *.c

archive: ${objects}
	zip -r component-${VERSION}.zip $<
```



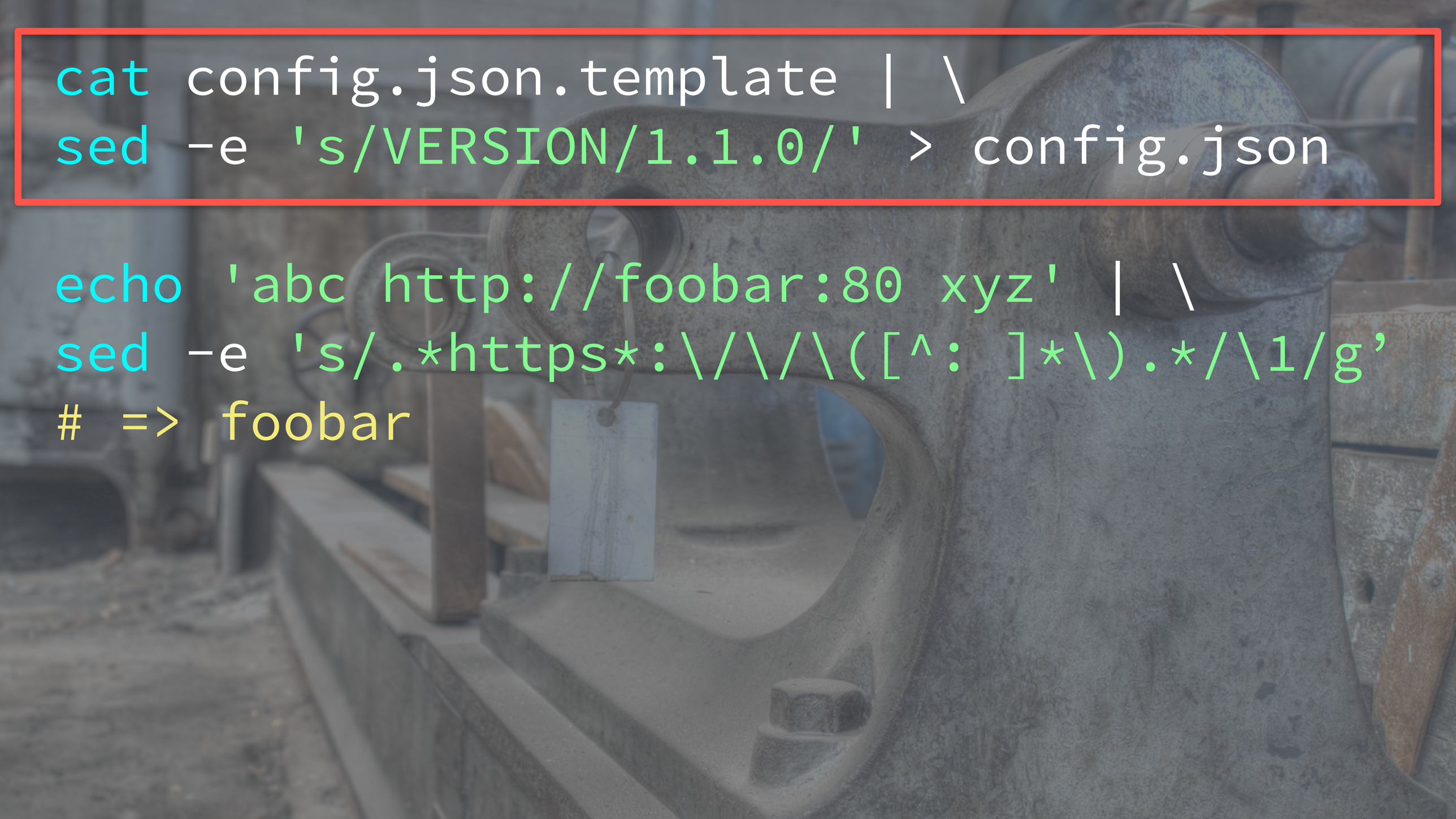

```
clean:
	@rm -f *.o *.zip
	@for c in ${components}; \
	  do rm -f $$c/*.o $$c/*.zip; done

validate:
	@validate.sh *.c

archive: ${objects}
	zip -r component-${VERSION}.zip $<
```


RANSOMES & RAPIER, LTD
MAKERS.
IPSWICH, ENGLAND.



- Conceptually simple, powerful
  - … if Quirky…
- Strikes a good balance of built-ins vs. user-control


sed
for editing



```
cat config.json.template | \
sed -e 's/VERSION/1.1.0/' > config.json
```

```
echo 'abc http://foobar:80 xyz' | \
sed -e 's/.*https*:\/\/\([^: ]*\).*/\1/g’
# => foobar
```

```
cat config.json.template | \
sed -e 's/VERSION/1.1.0/' > config.json
```

```
echo 'abc http://foobar:80 xyz' | \
sed -e 's/.*https*:\/\/\([^: ]*\).*/\1/g’
# => foobar
```

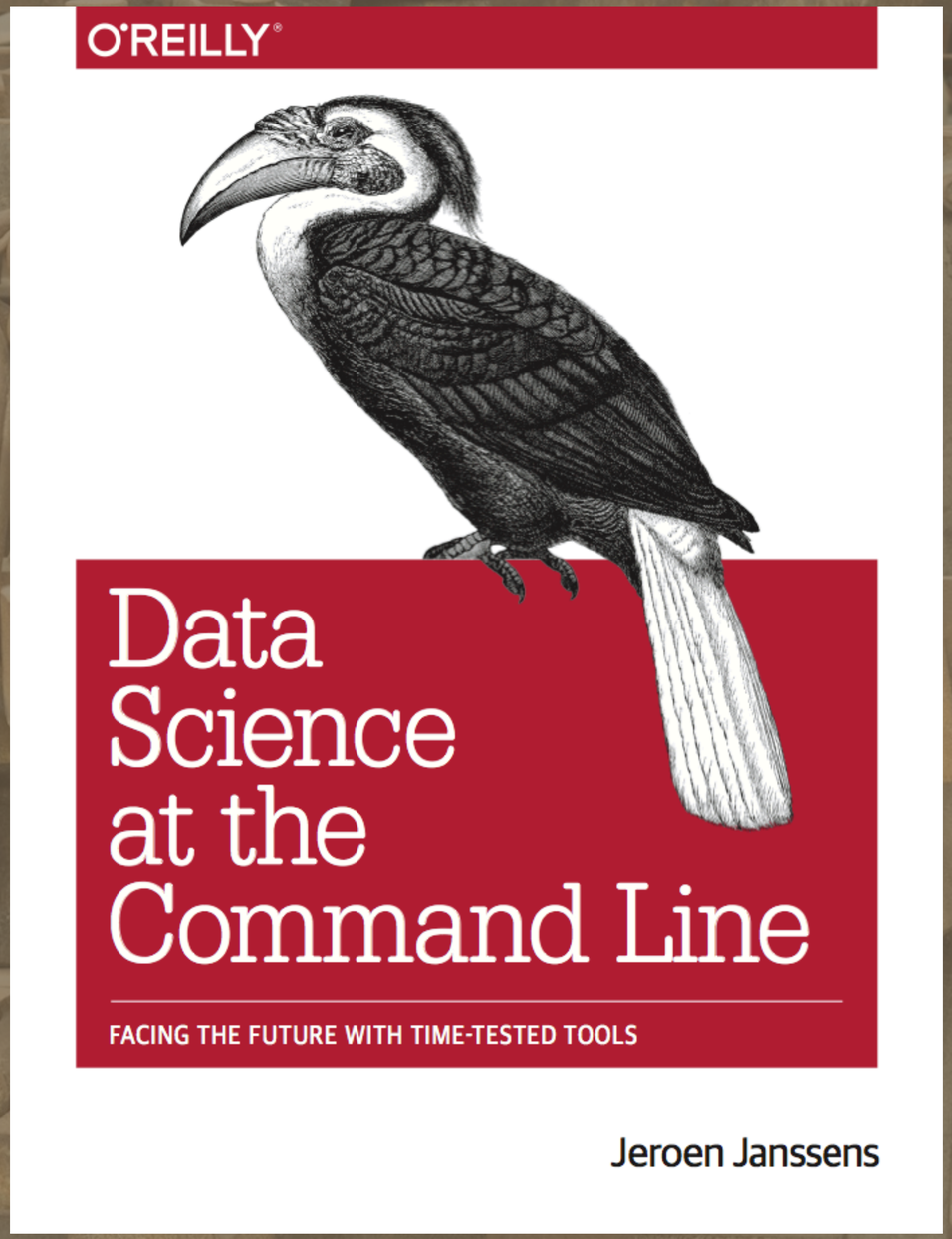

grep
for editing

```
grep -rino --max-count=3 --extended \
  'https?://[^ ]*lightbend.com' \
  src/main/paradox
# …/user-guide/running-apps/index.md:
1166:https://www.lightbend.com
# …/management-guide/index.md:
393:https://developer.lightbend.com
# …/installation/index.md:58:http://
developer.lightbend.com
```

UNIX Philosophy Today

for data

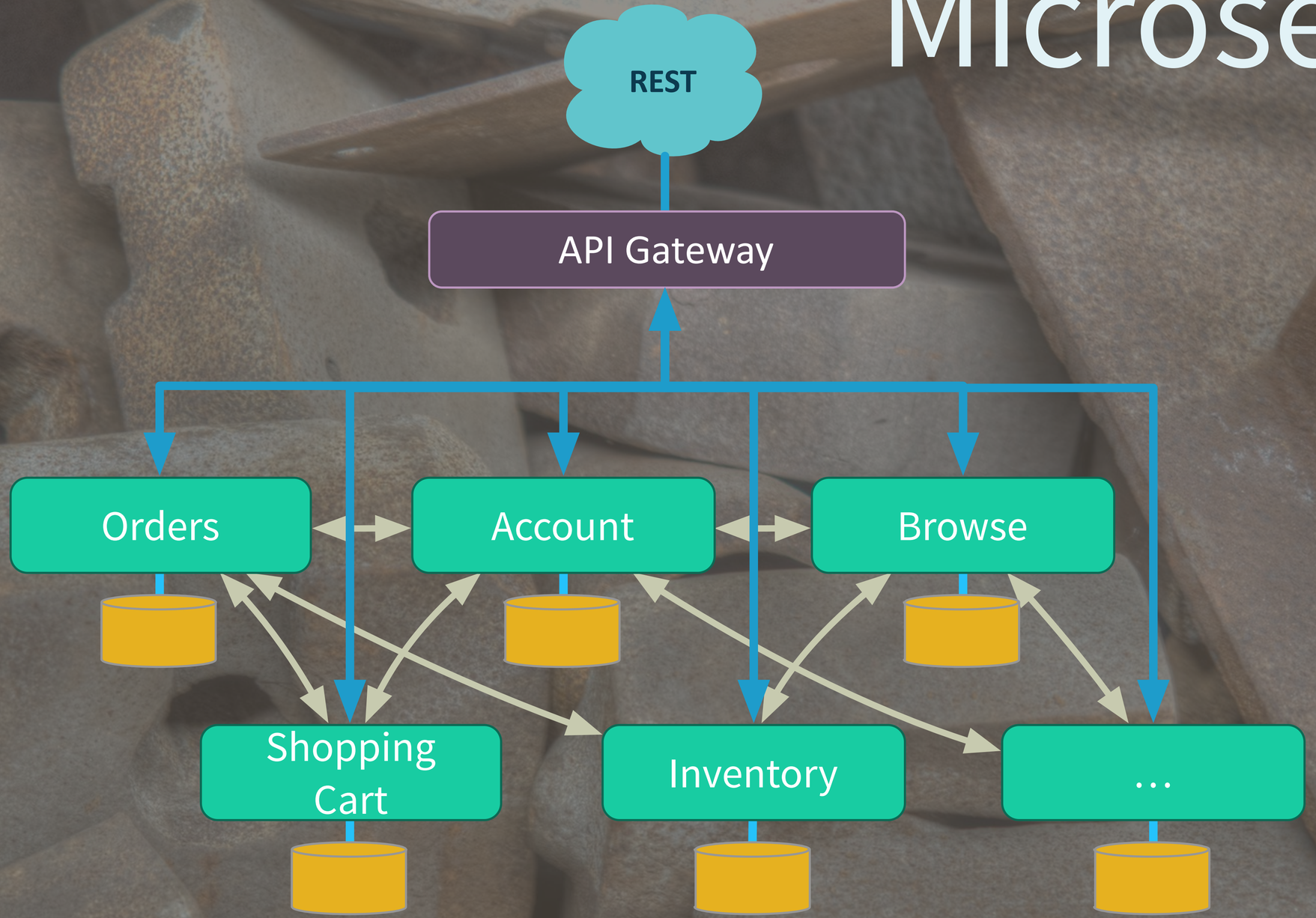
datascienceatthecommandline.com



Spark

```
spark.textFile("data/…")
  .map { line =>
    val array = line .split("\t", 2)
    (array(0), array(1))
  }
  .flatMap {
    case (path, text) =>
      text.split("\\W+") map {(_, path)}
  }
  .map { … } …
```

Microservices



Questions?

Dean Wampler, Ph.D.
dean@lightbend.com
@deanwampler
polyglotprogramming.com/talks
lightbend.com/fast-data-platform

Lightbend