Tracing Software Build Processes to Uncover License Compliance Inconsistencies

Sander van der Burg

TU Delft

Delft University of Technology

Eelco Dolstra

LogicBlox

Shane McIntosh

Queen's University

Julius Davies

UBC

Daniel M. Germán

University of Victoria

Armijn Hemel

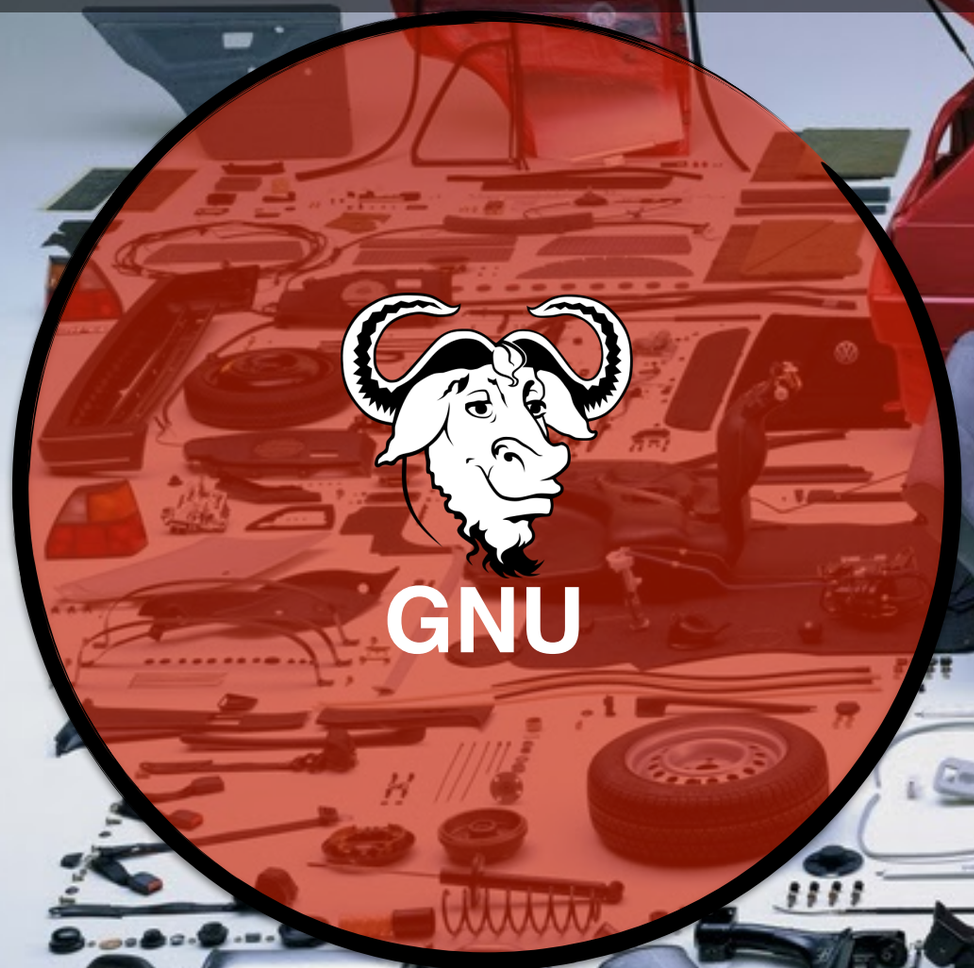
Tjaldur

Software Governance Solutions

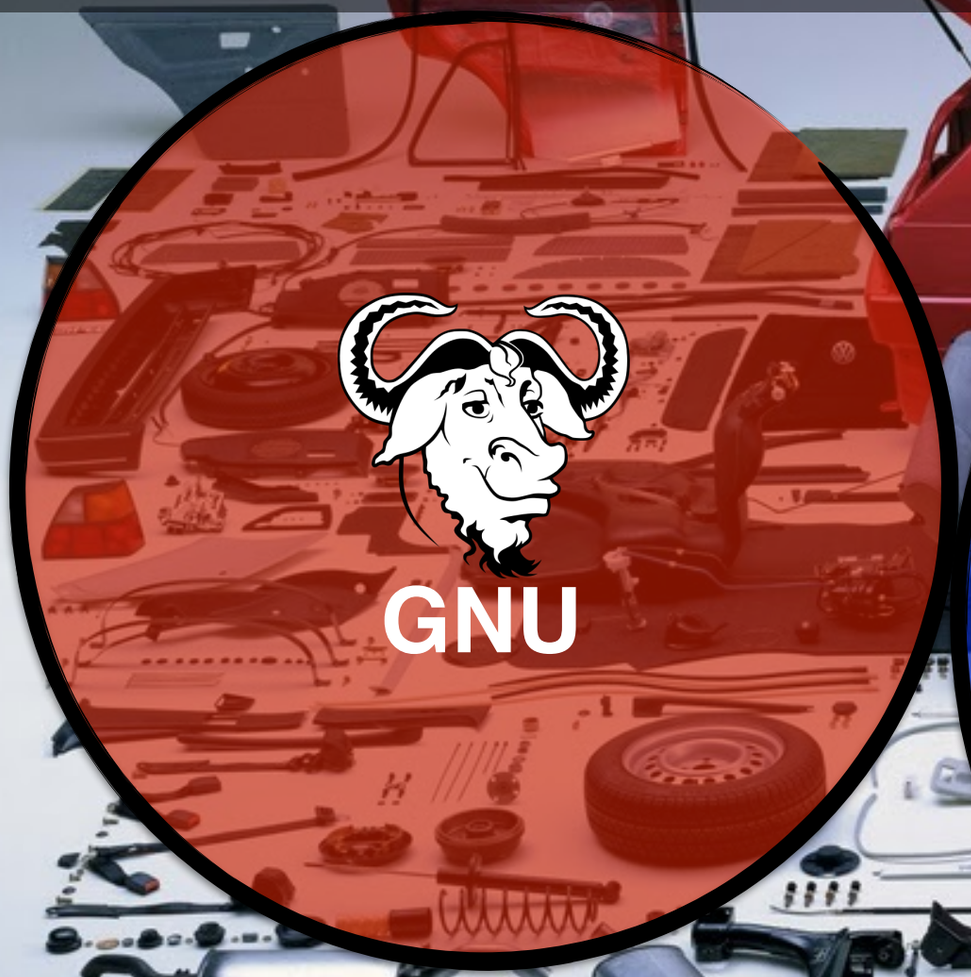
@shane_mcintosh

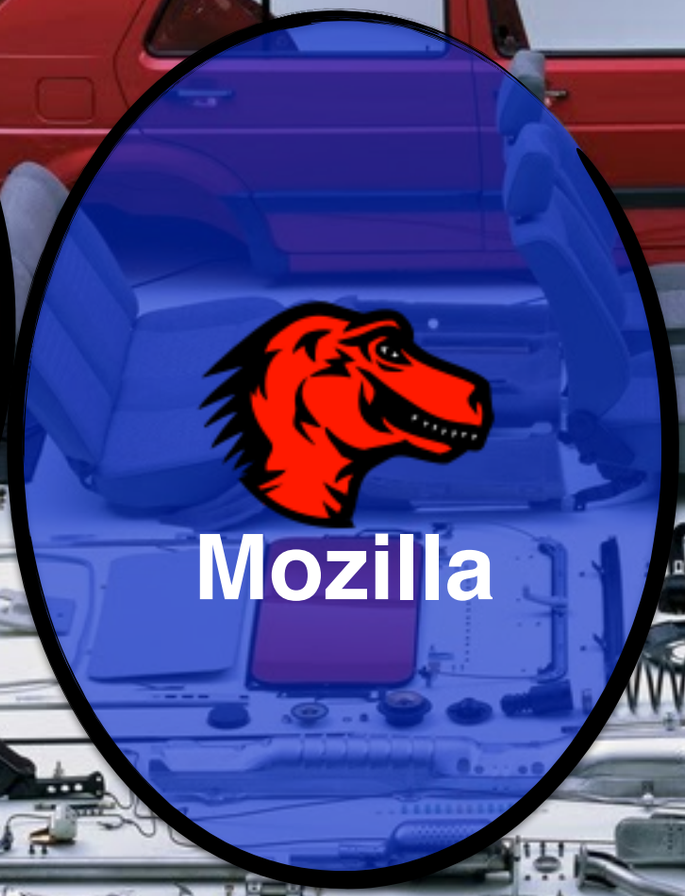
Software reuse enables rapid development of new applications

Software reuse enables rapid development of new applications

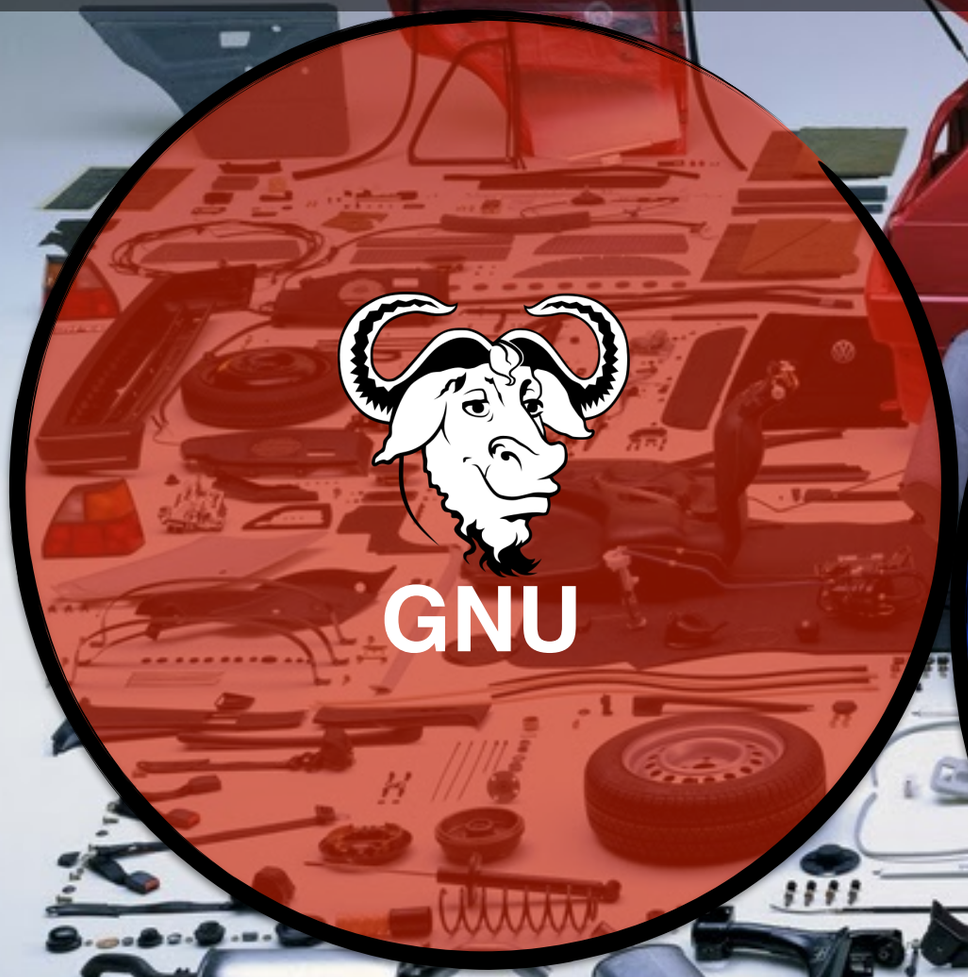


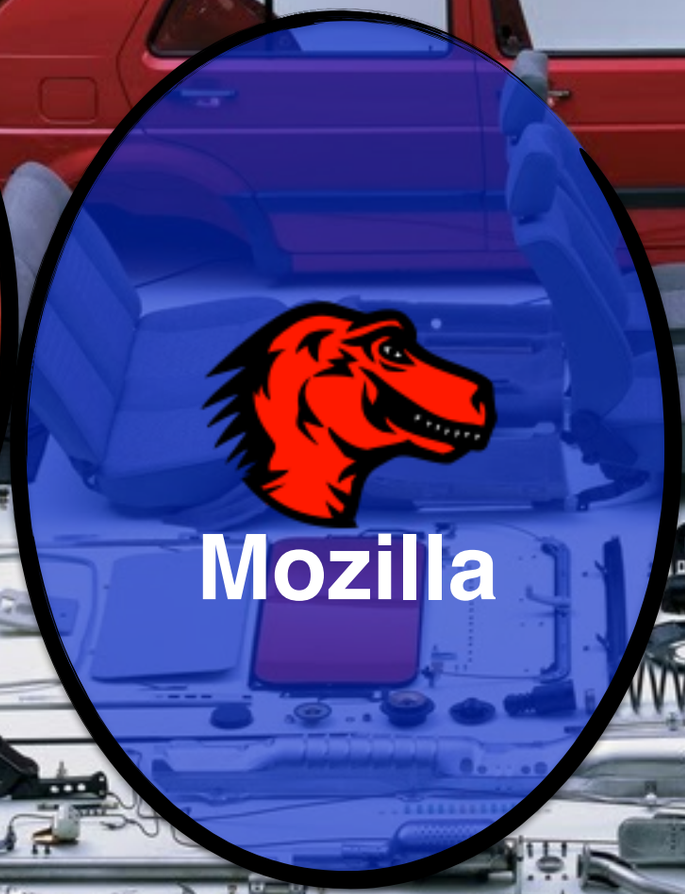
Software reuse enables rapid development of new applications



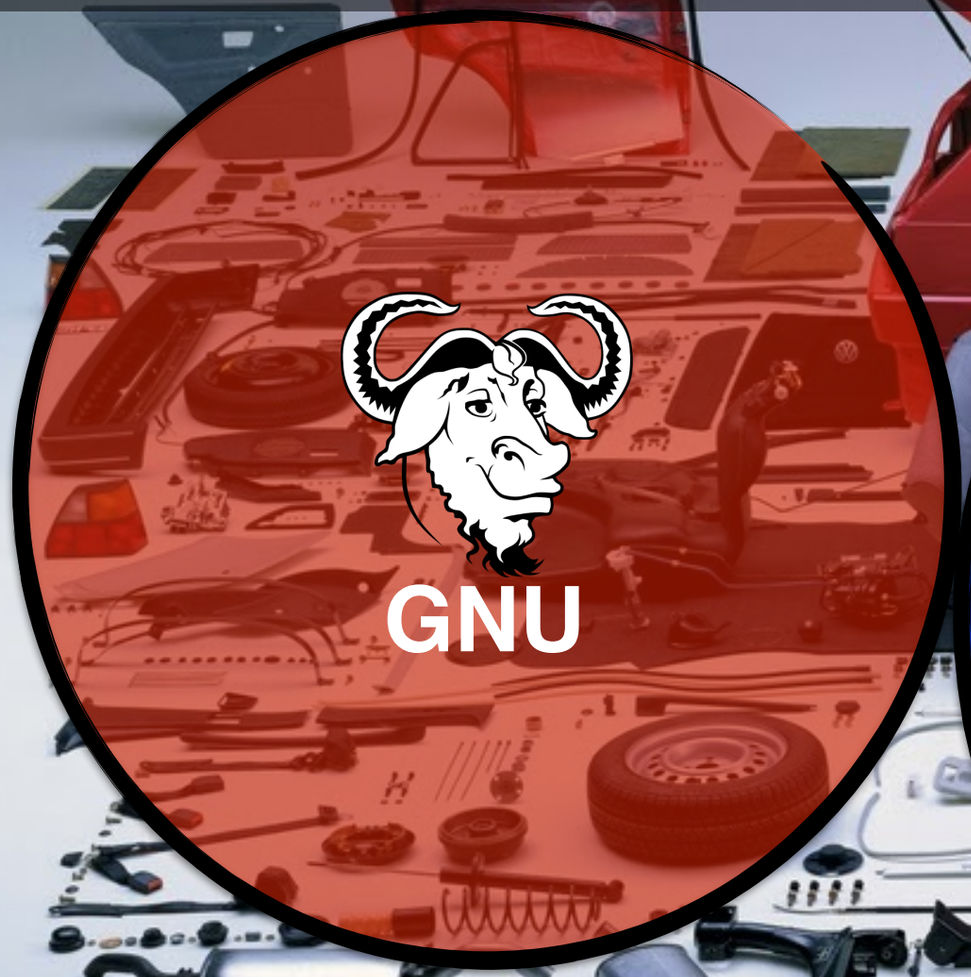


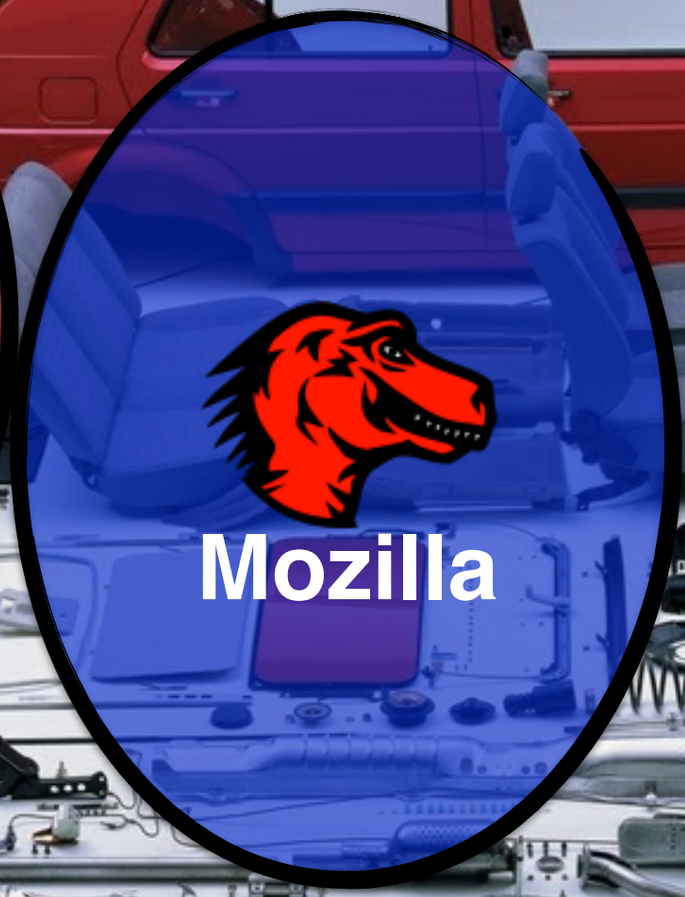
Software reuse enables rapid development of new applications



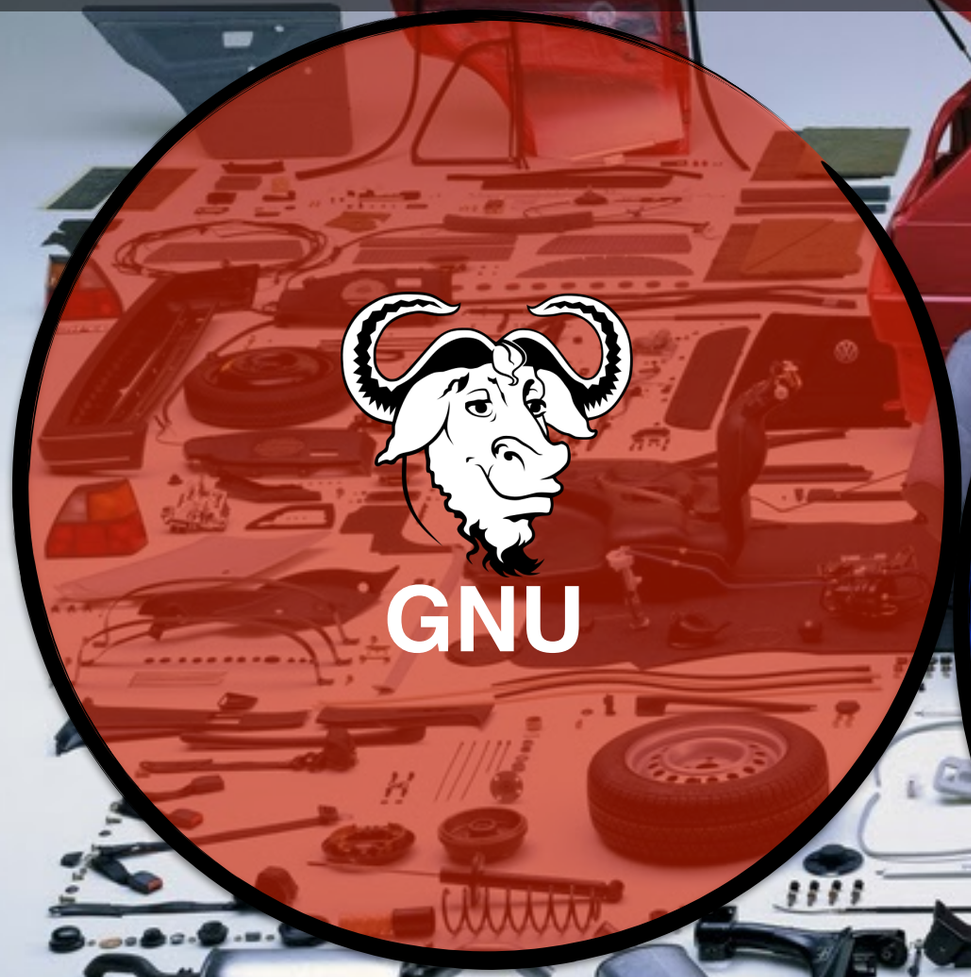


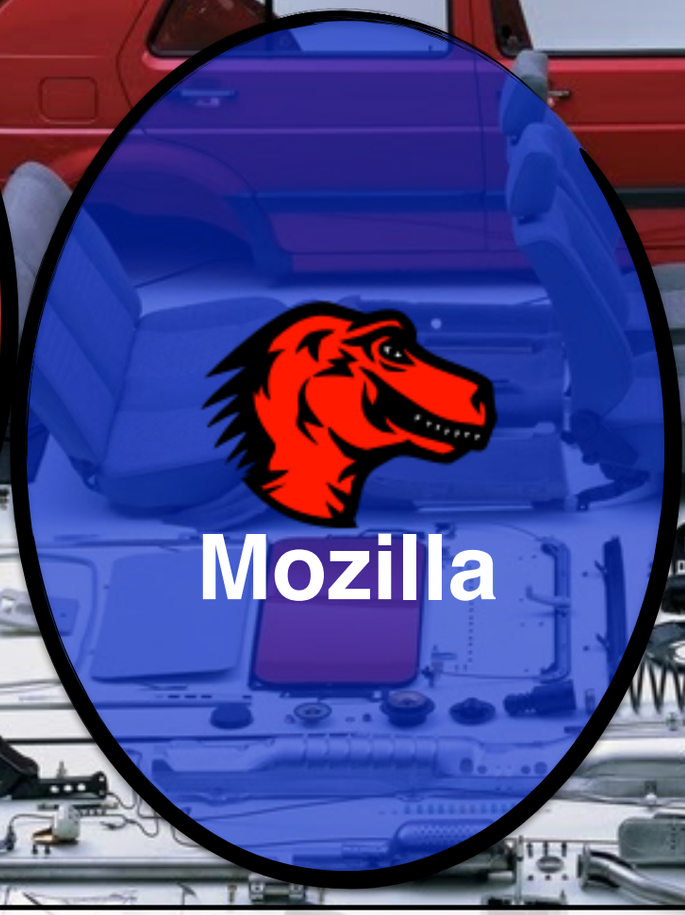
Reusable components are released under different license terms



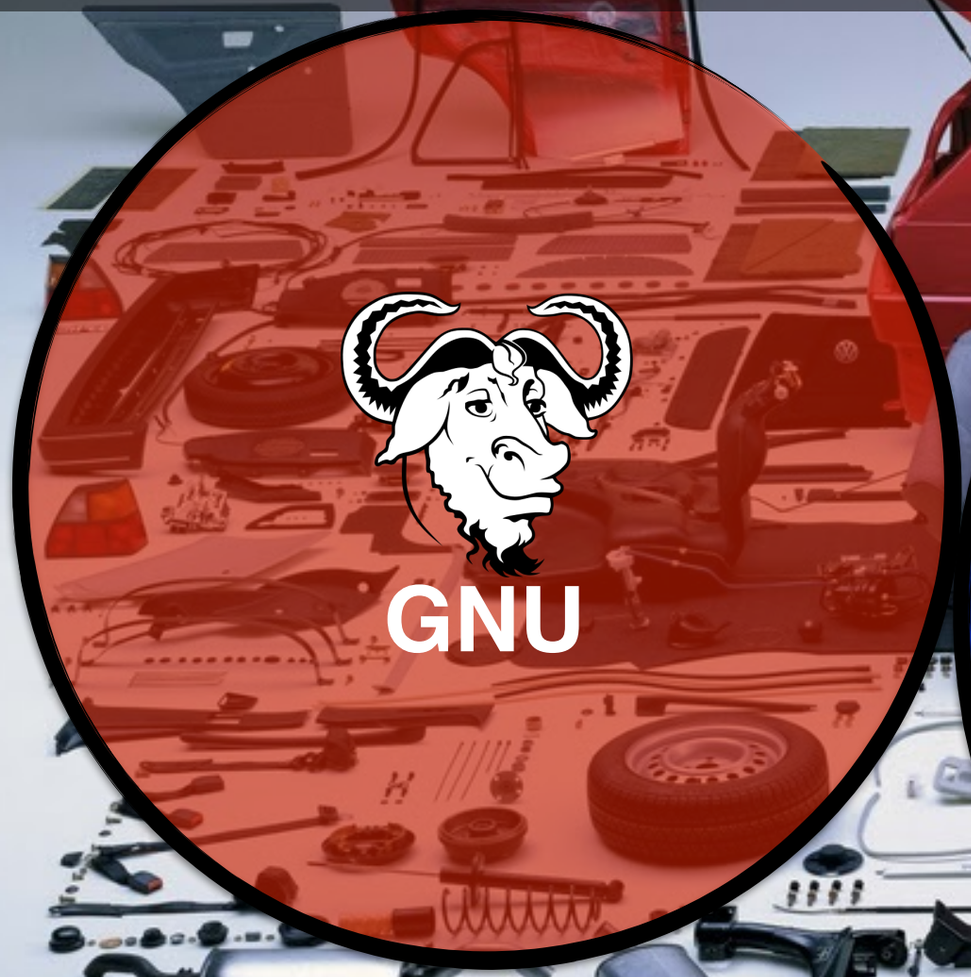


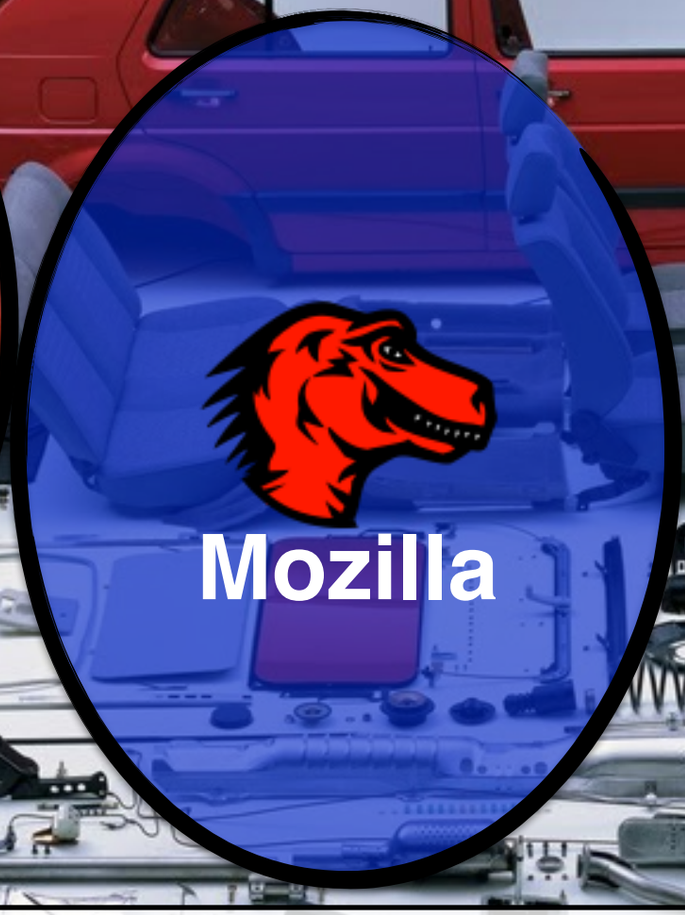
Reusable components are released under different license terms

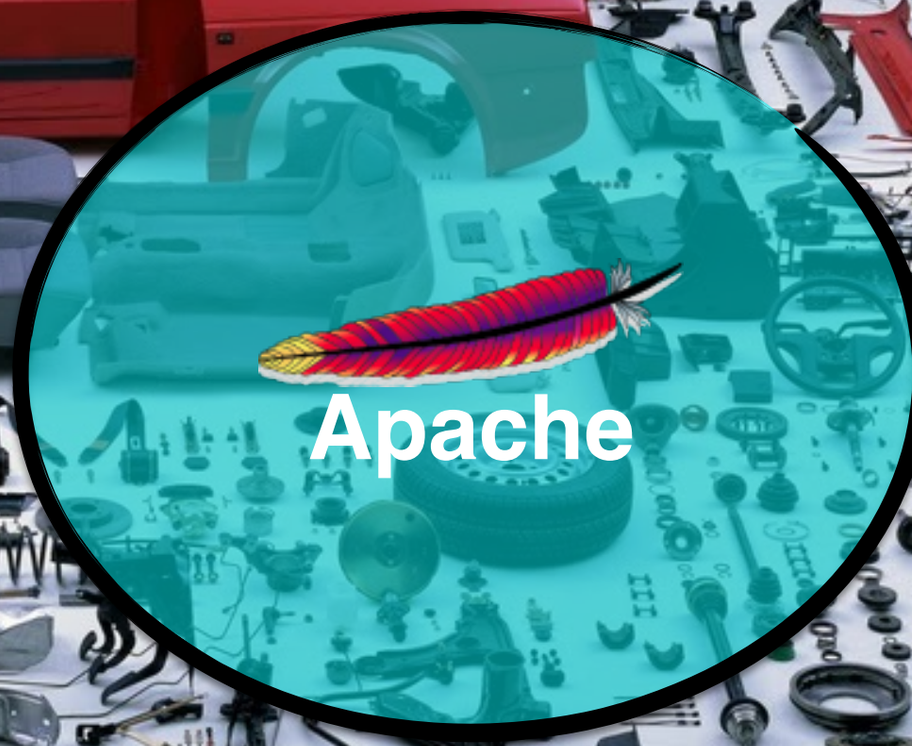




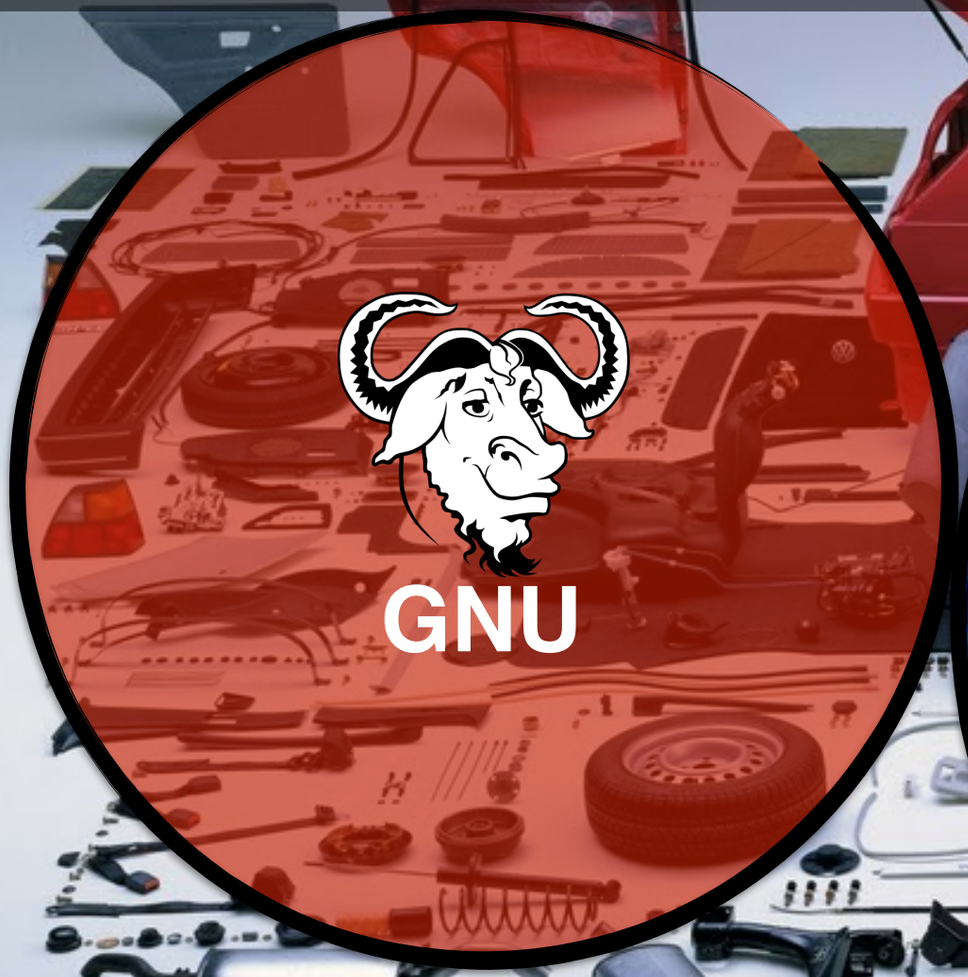
Reusable components are released under different license terms

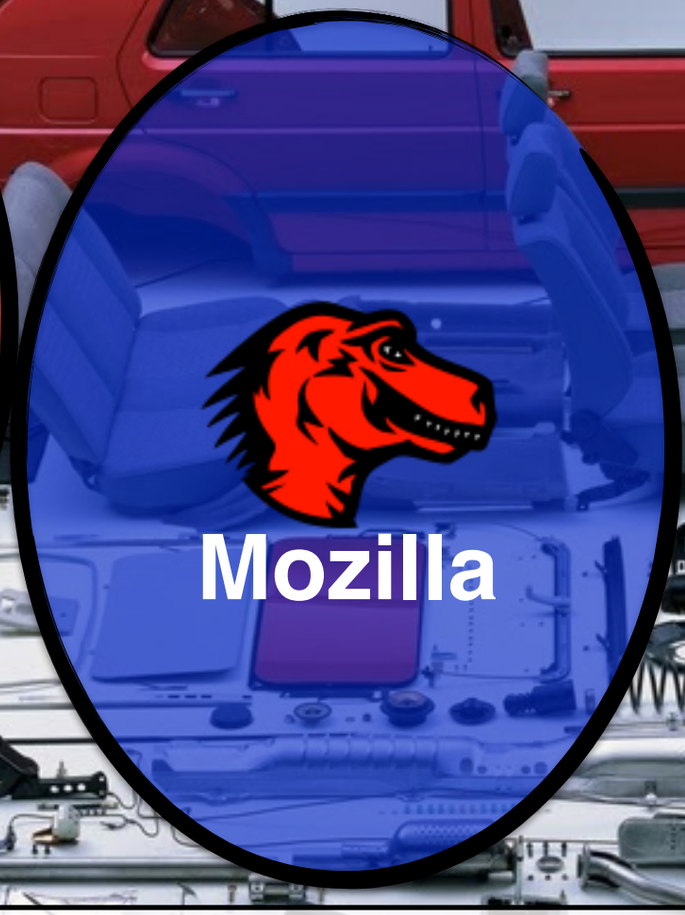


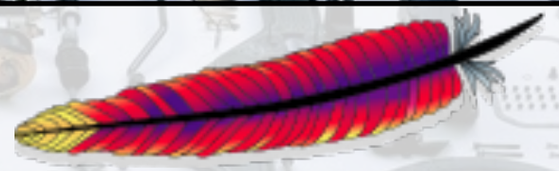




Reusable components are released under different license terms







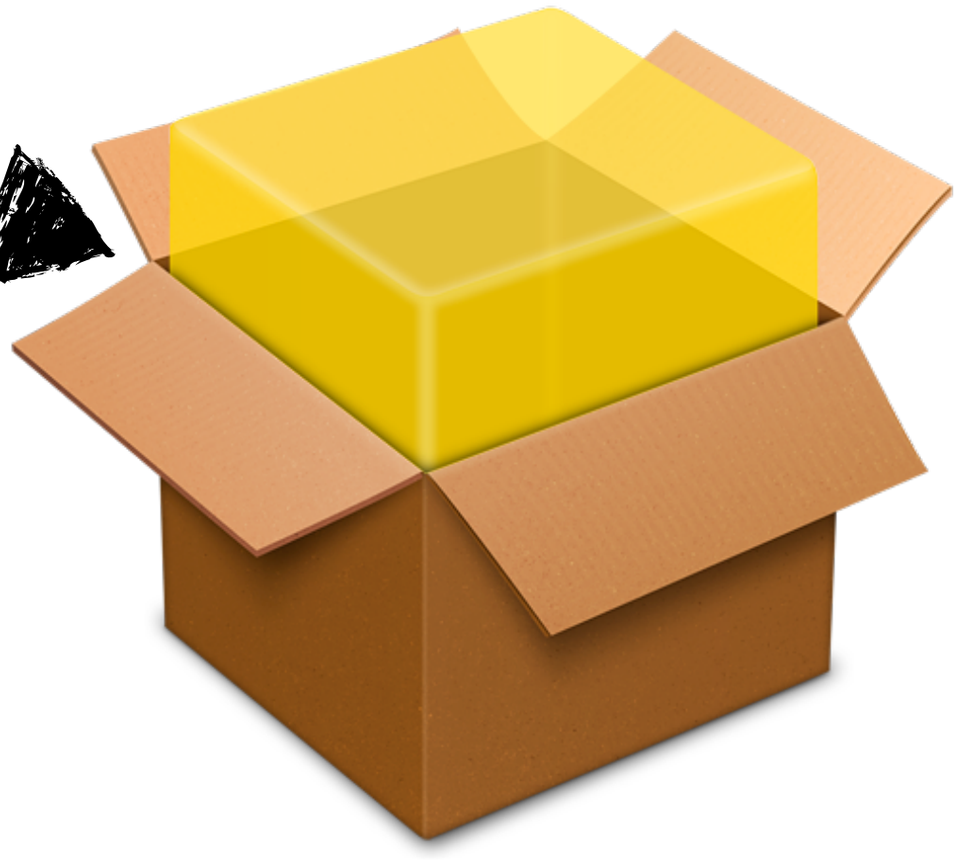
Apache Public License

Reuse puts legal constraints on how client systems can be distributed

External component

Used by

Client system



Reuse puts legal constraints on how client systems can be distributed

External component

Used by

Client system

Failure to comply with license terms can lead to costly legal issues

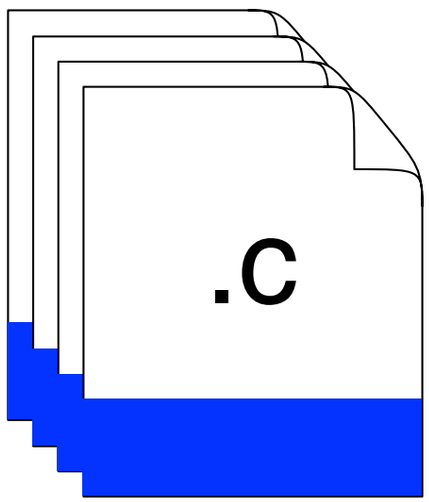
Failure to comply with license terms can lead to costly legal issues

Failure to comply with license terms can lead to costly legal issues

Failure to comply with license terms can lead to costly legal issues

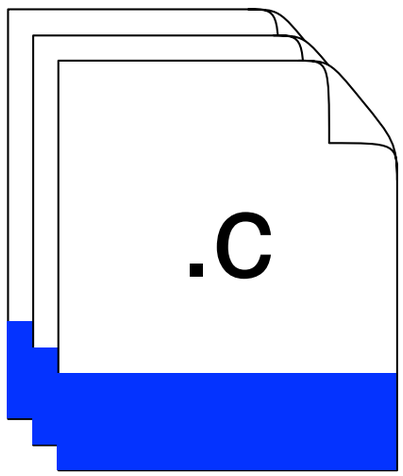
frdm Software Freedom Law Center

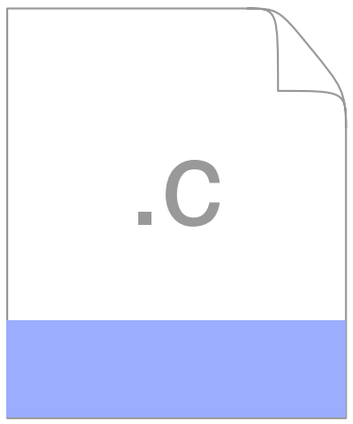
Ensuring license compliance with reused components



Which source files are enabled?

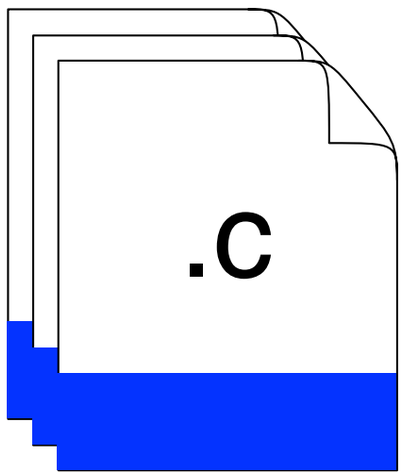
Ensuring license compliance with reused components

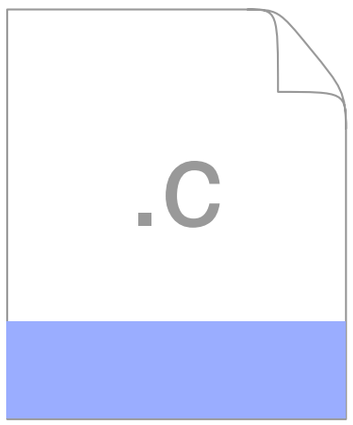




Which source files are enabled?

Ensuring license compliance with reused components

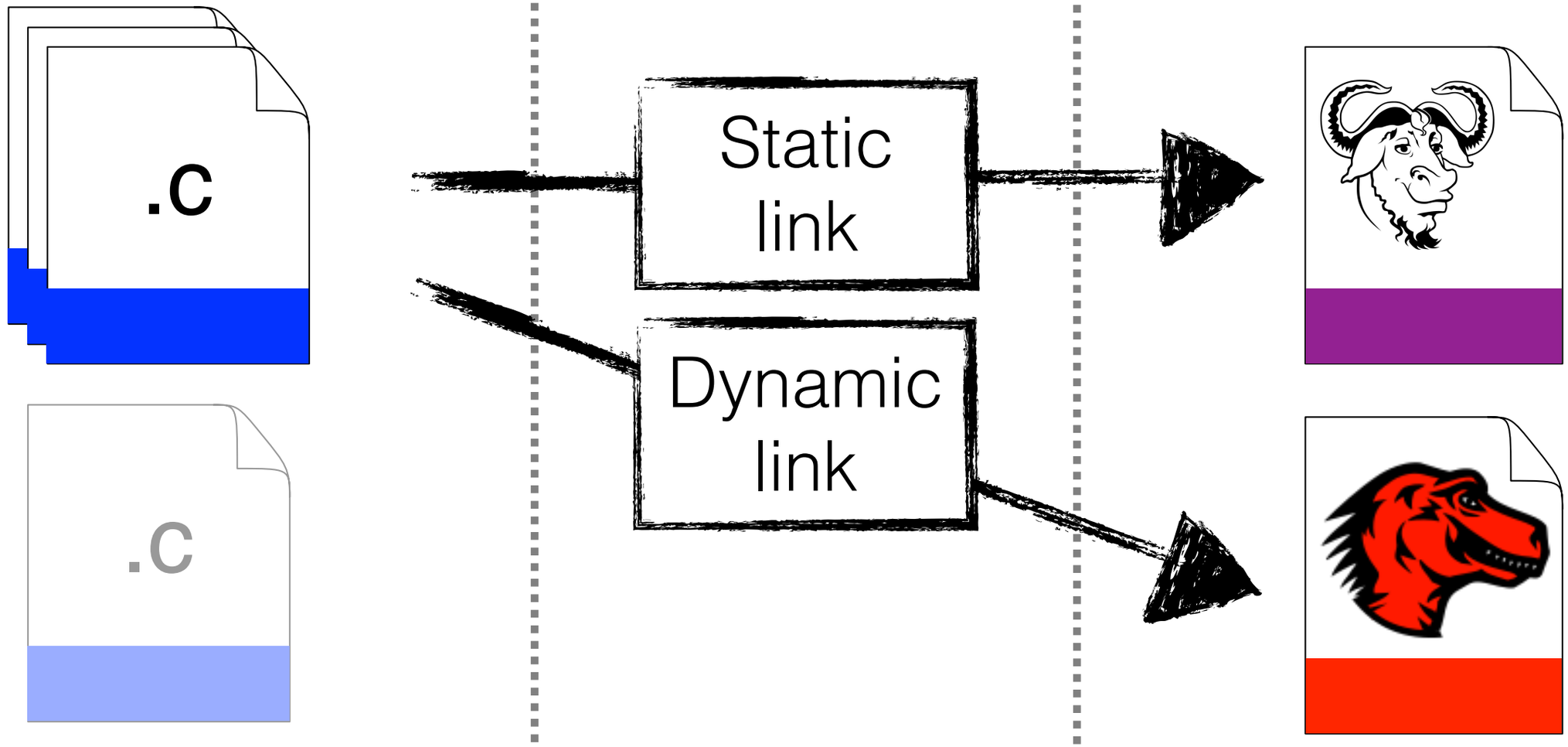




Which source files are enabled?

Which components are used?

Ensuring license compliance with reused components

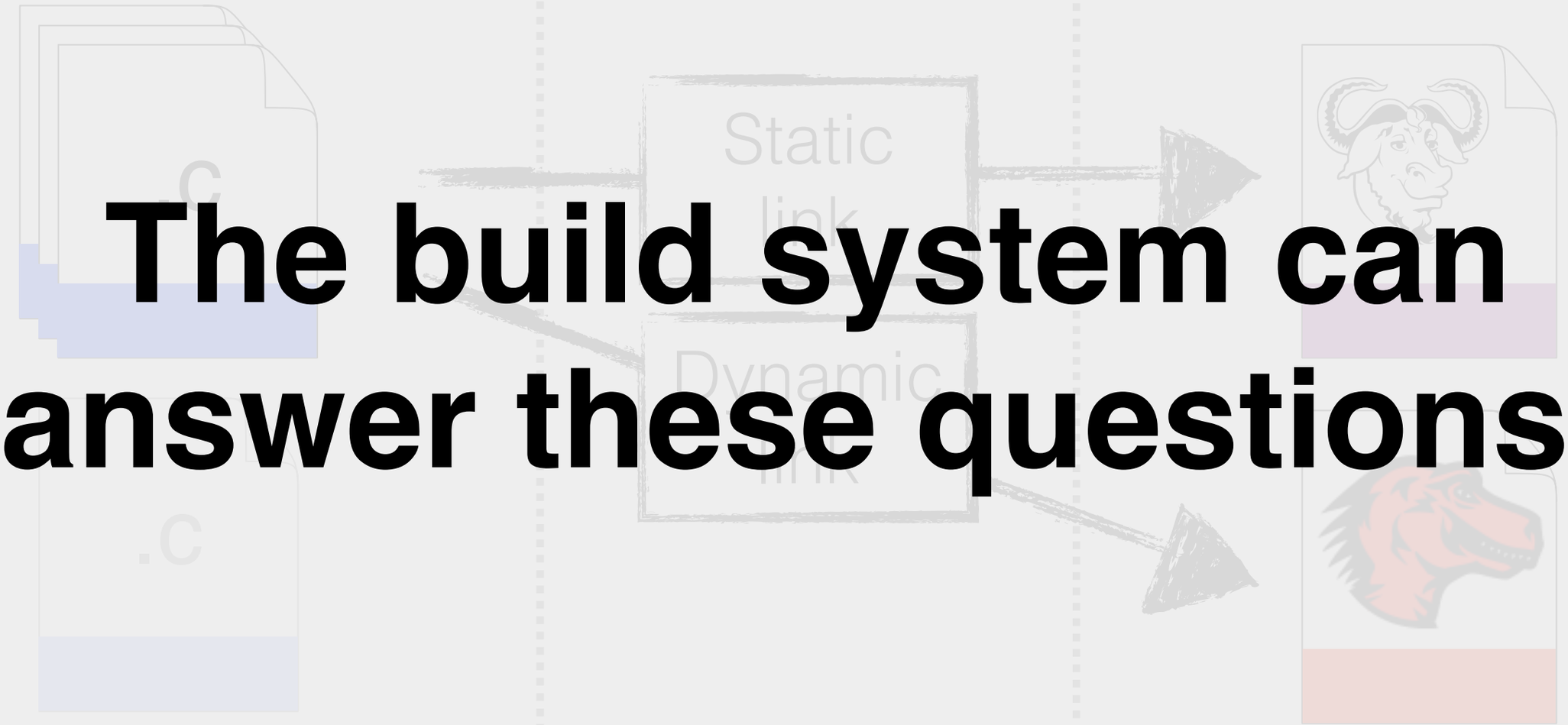


Which source files are enabled?

How are they combined?

Which components are used?

Ensuring license compliance with reused components



The build system can answer these questions!

Which source files are enabled?

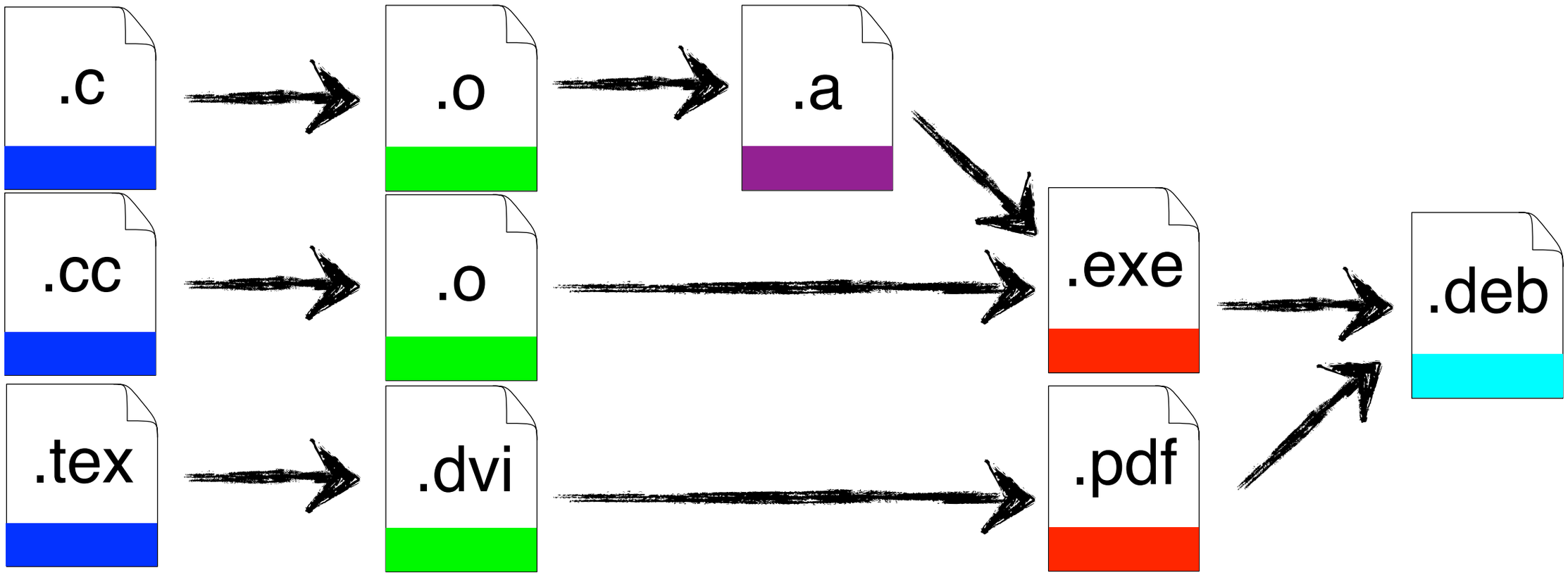
How are they combined?

Which components are used?

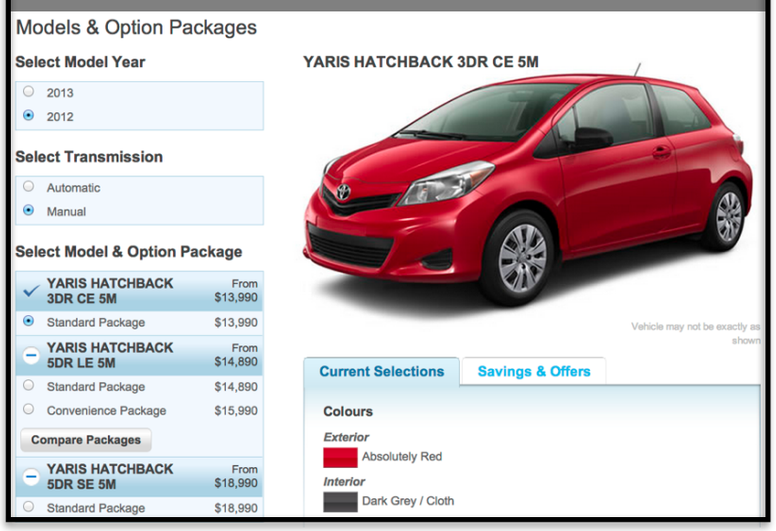
What is a build system?

What is a build system?

Build systems describe how sources are translated into deliverables



Step 1 - Configuration



Step 2 - Construction

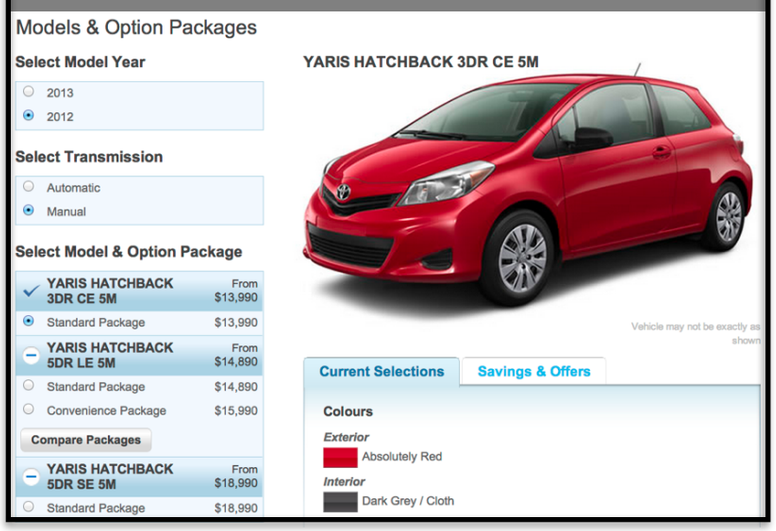
Step 3 - Certification

Step 4 - Packaging

Step 5 - Deployment

Focus of this paper

Step 1 - Configuration



Step 2 - Construction

Step 3 - Certification

Step 4 - Packaging

Step 5 - Deployment

Step 1 - Configuration

Models & Option Packages

Select Model Year

- ○ 2013
- ◉ 2012

Select Transmission

- ○ Automatic
- ◉ Manual

Select Model & Option Package

✓	**YARIS HATCHBACK 3DR CE 5M**	From $13,990
◉	Standard Package	$13,990
−	**YARIS HATCHBACK 5DR LE 5M**	From $14,890
○	Standard Package	$14,890
○	Convenience Package	$15,990
	Compare Packages	
−	**YARIS HATCHBACK 5DR SE 5M**	From $18,990
○	Standard Package	$18,990

YARIS HATCHBACK 3DR CE 5M

Vehicle may not be exactly as shown

Current Selections | Savings & Offers

Colours

Exterior
Absolutely Red

Interior
Dark Grey / Cloth

Step 2 - Construction

Incompleteness of build specs makes license compliance assessment difficult

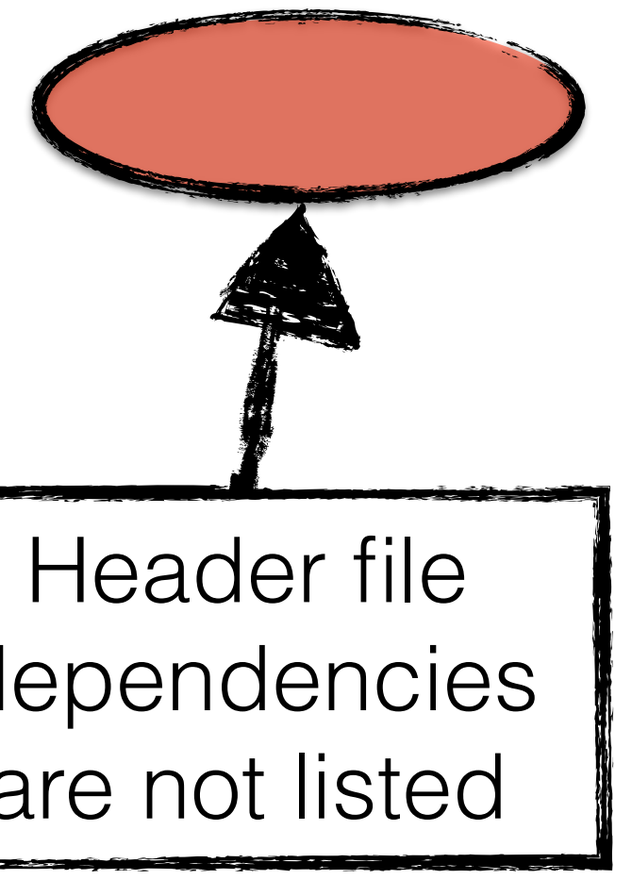
```
patchelf.o: patchelf.cc
  g++ -c patchelf.cc

patchelf: patchelf.o
  g++ patchelf.o -o patchelf

install: patchelf
  install patchelf /usr/bin/
```

Incompleteness of build specs makes license compliance assessment difficult

```
patchelf.o: patchelf.cc
  g++ -c patchelf.cc

patchelf: patchelf.o
  g++ patchelf.o -o patchelf

install: patchelf
  install patchelf /usr/bin/
```

Incompleteness of build specs makes license compliance assessment difficult

```
patchelf.o: patchelf.cc
  g++ -c patchelf.cc

patchelf: patchelf.o
  g++ patchelf.o -o pat

install: patchelf
  install patchelf /usr/bin/
```



Incompleteness of build specs makes license compliance assessment difficult

elf.h

patchelf.cc

patchelf.o

Dependencies

Extracted

Missing

Incompleteness of build specs makes license compliance assessment difficult

Dependencies

Extracted

Missing

Incompleteness of build specs makes license compliance assessment difficult

Dependencies

Extracted

Missing

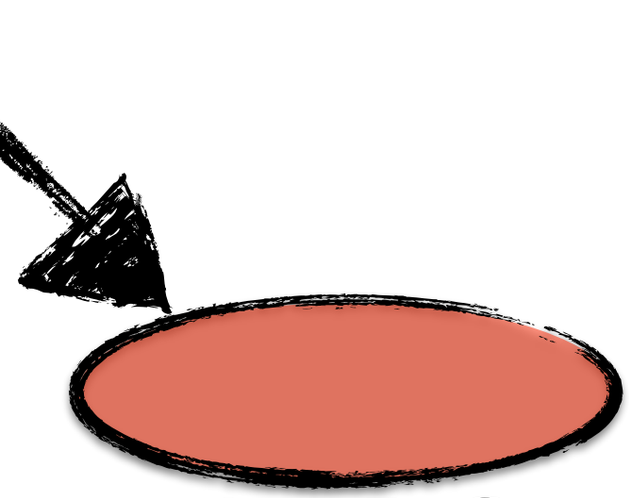
Incompleteness of build specs makes license compliance assessment difficult

```
patchelf.o: patchelf.cc
  g++ -c patchelf.cc

patchelf: patchelf.o
  g++ patchelf.o -o patchelf

install: patchelf
  install patchelf /usr/bin/
```

Incompleteness of build specs makes license compliance assessment difficult

```
patchelf.
  g++ -c
```

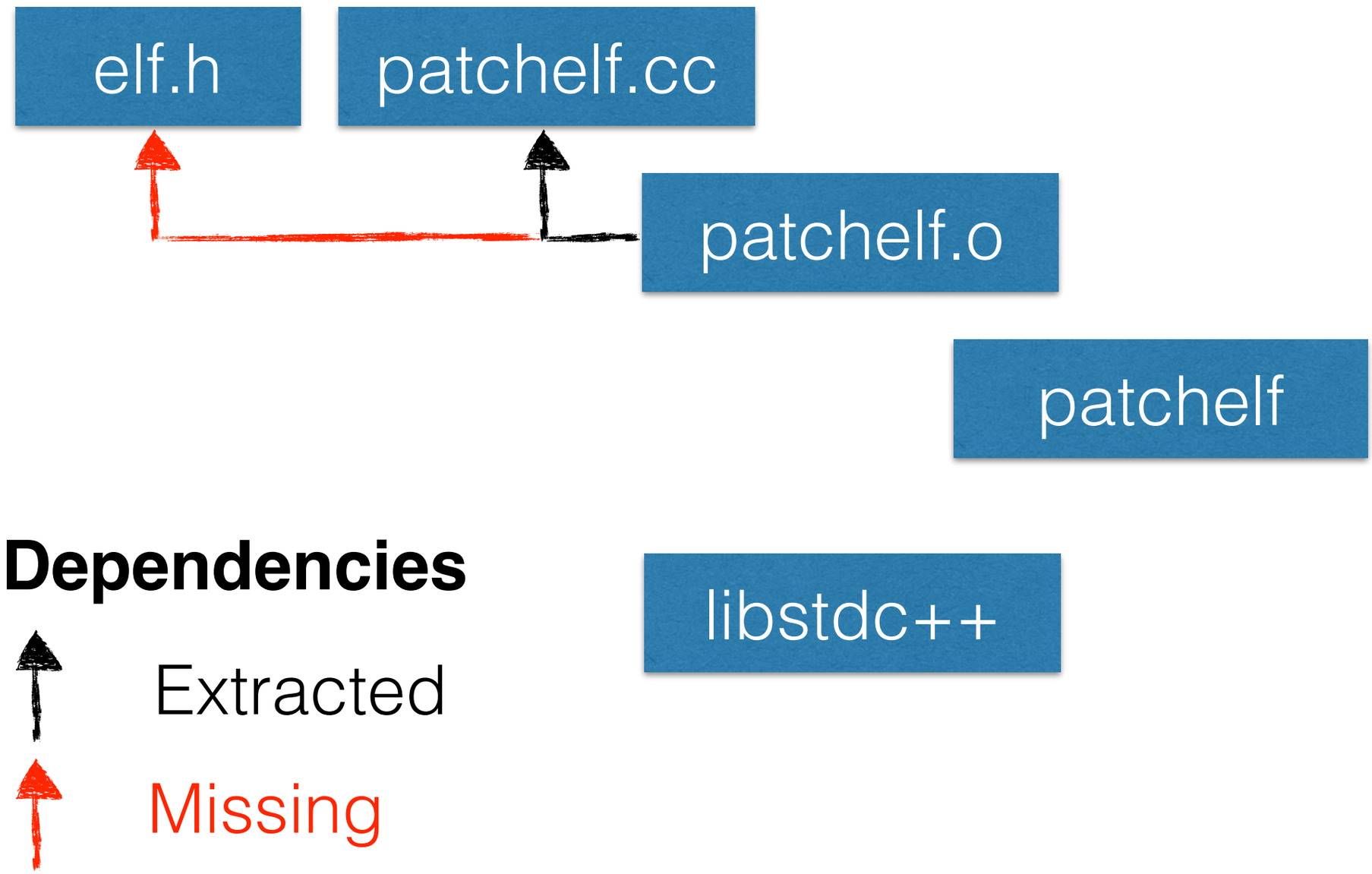
External library dependencies are not listed



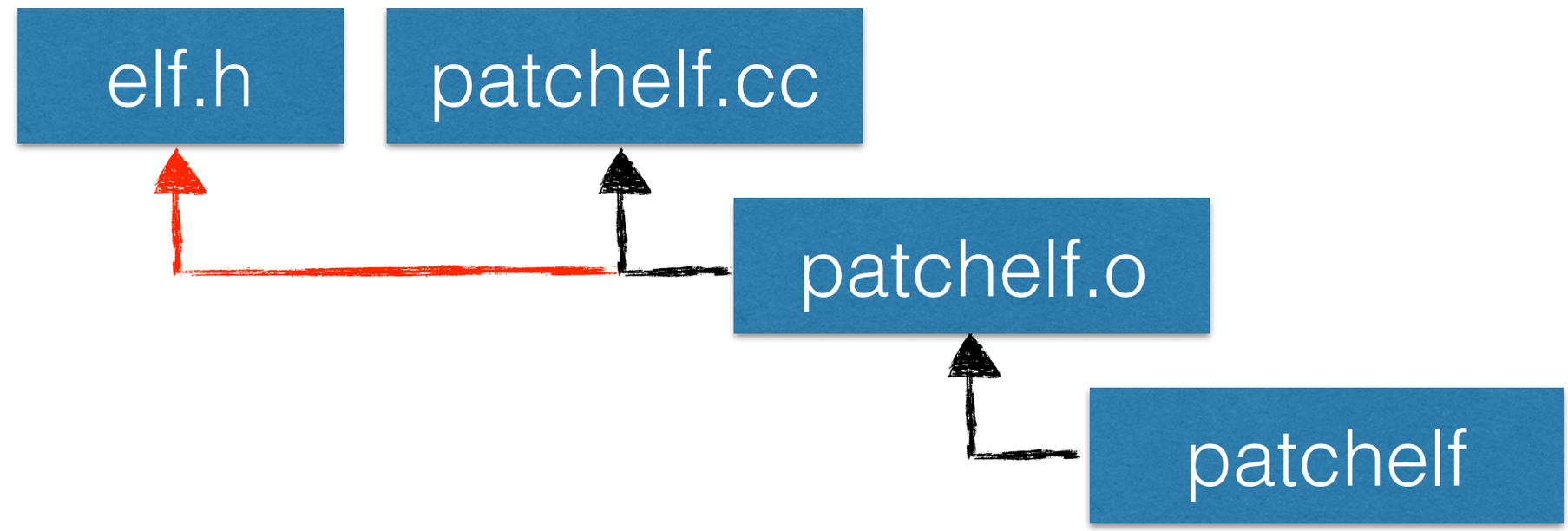
```
f.cc
c
```

```
patchelf: patchelf.o
  g++ patchelf.o -o patchelf

install: patchelf
  install patchelf /usr/bin/
```

Incompleteness of build specs makes license compliance assessment difficult



Incompleteness of build specs makes license compliance assessment difficult



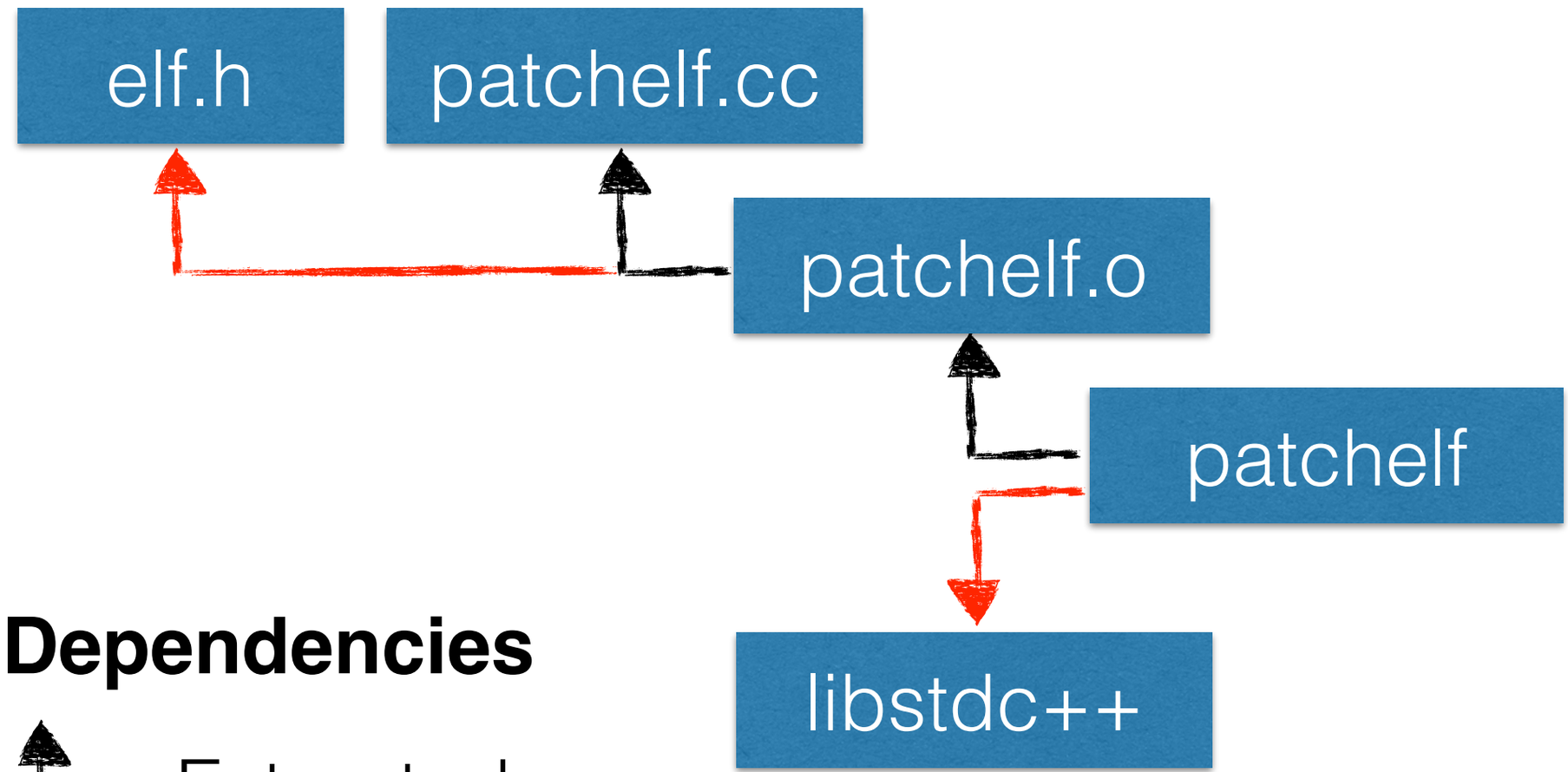
Dependencies

Extracted

Missing

libstdc++

Incompleteness of build specs makes license compliance assessment difficult



Incompleteness of build specs makes license compliance assessment difficult

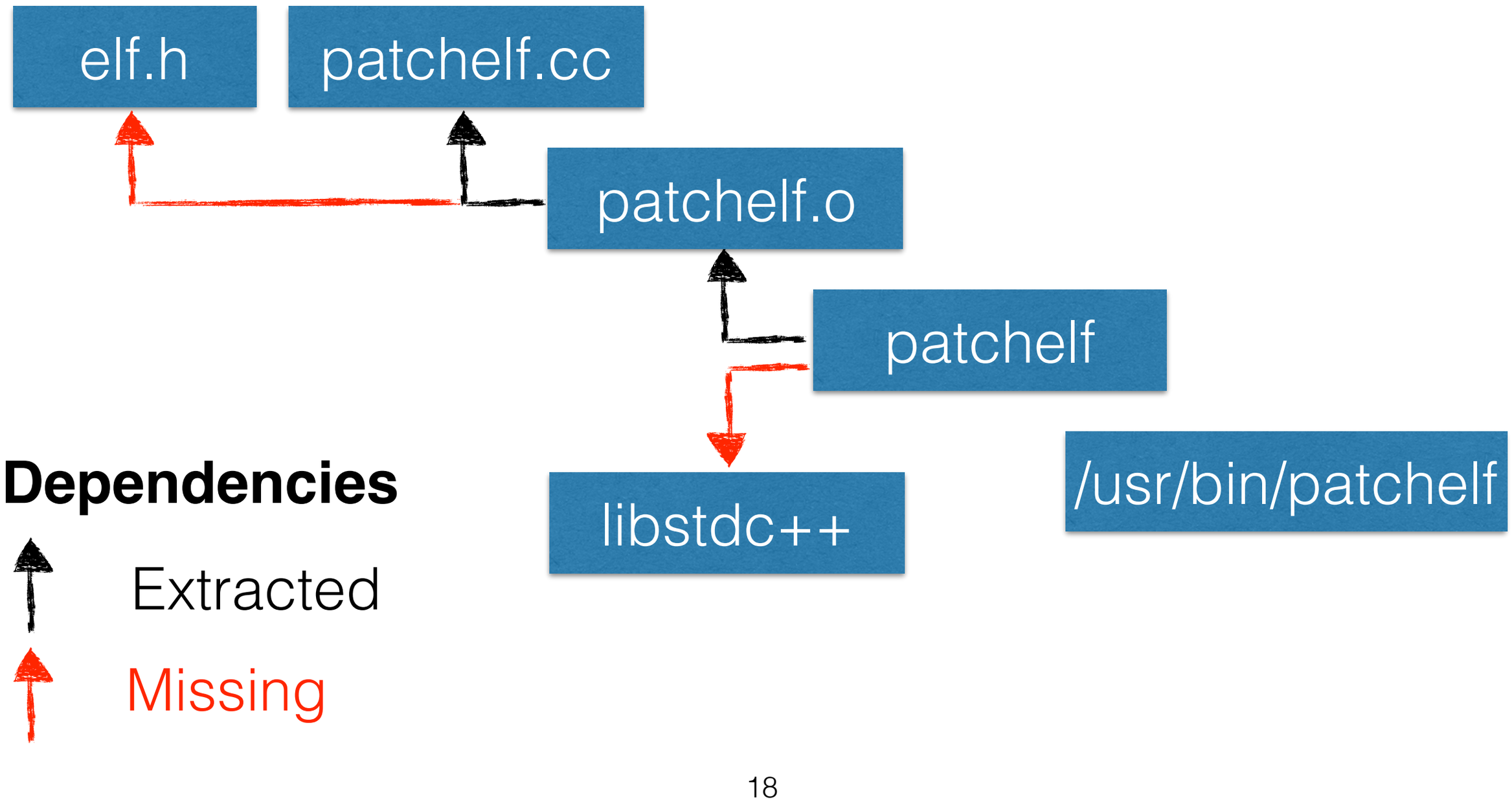
```
patchelf.o: patchelf.cc
  g++ -c patchelf.cc

patchelf: patchelf.o
  g++ patchelf.o -o patchelf

install: patchelf
  install patchelf /usr/bin/
```

Incompleteness of build specs makes license compliance assessment difficult

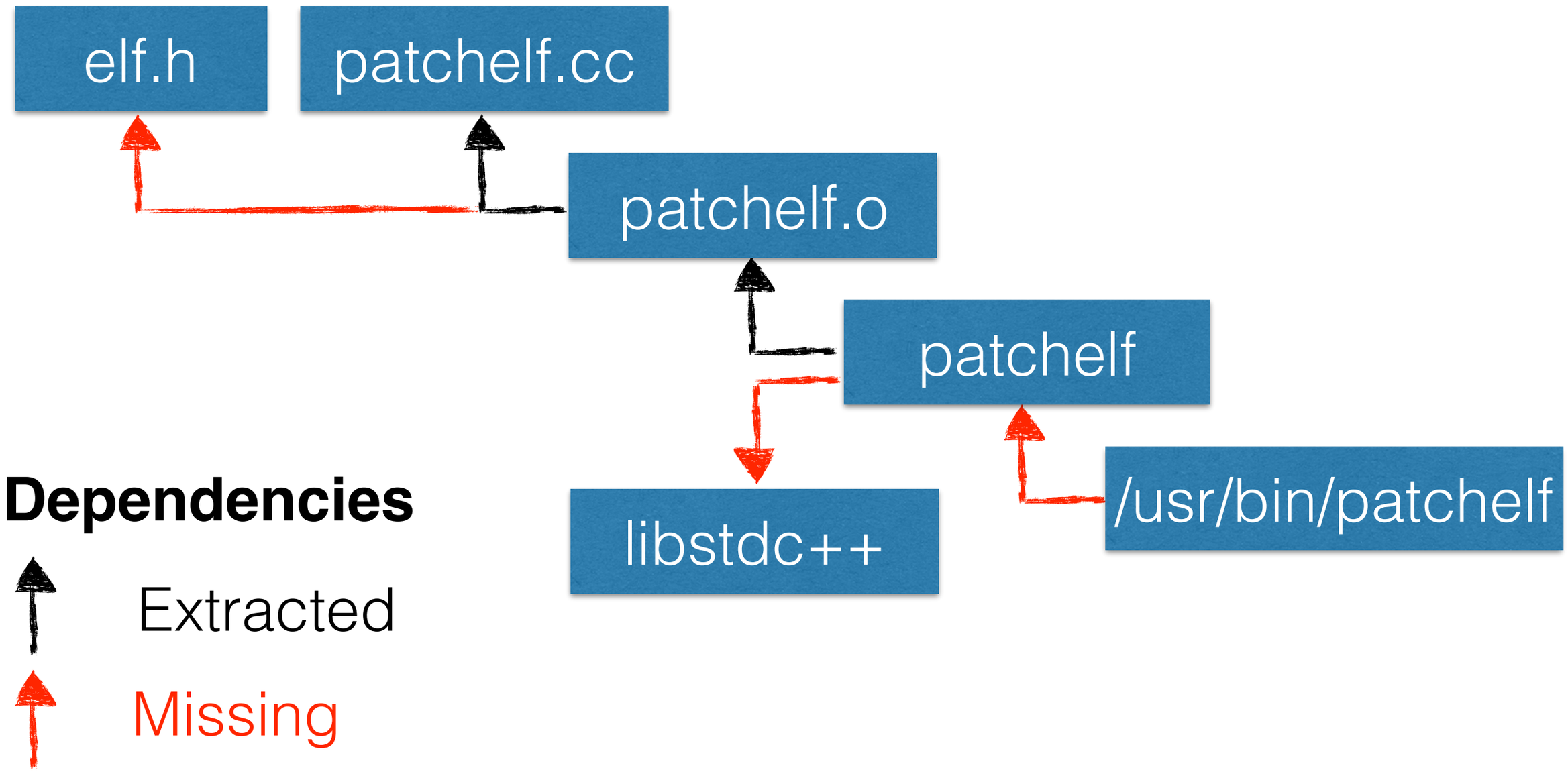
```
patchelf.o: patchelf.cc
	g++ -c patchelf.cc

patchelf: patchelf.o
	g++ patchelf.o -o patchelf

install: patchelf
	install patchelf /usr/bin/
```

Hidden relationship between patchelf and /usr/bin/patchelf

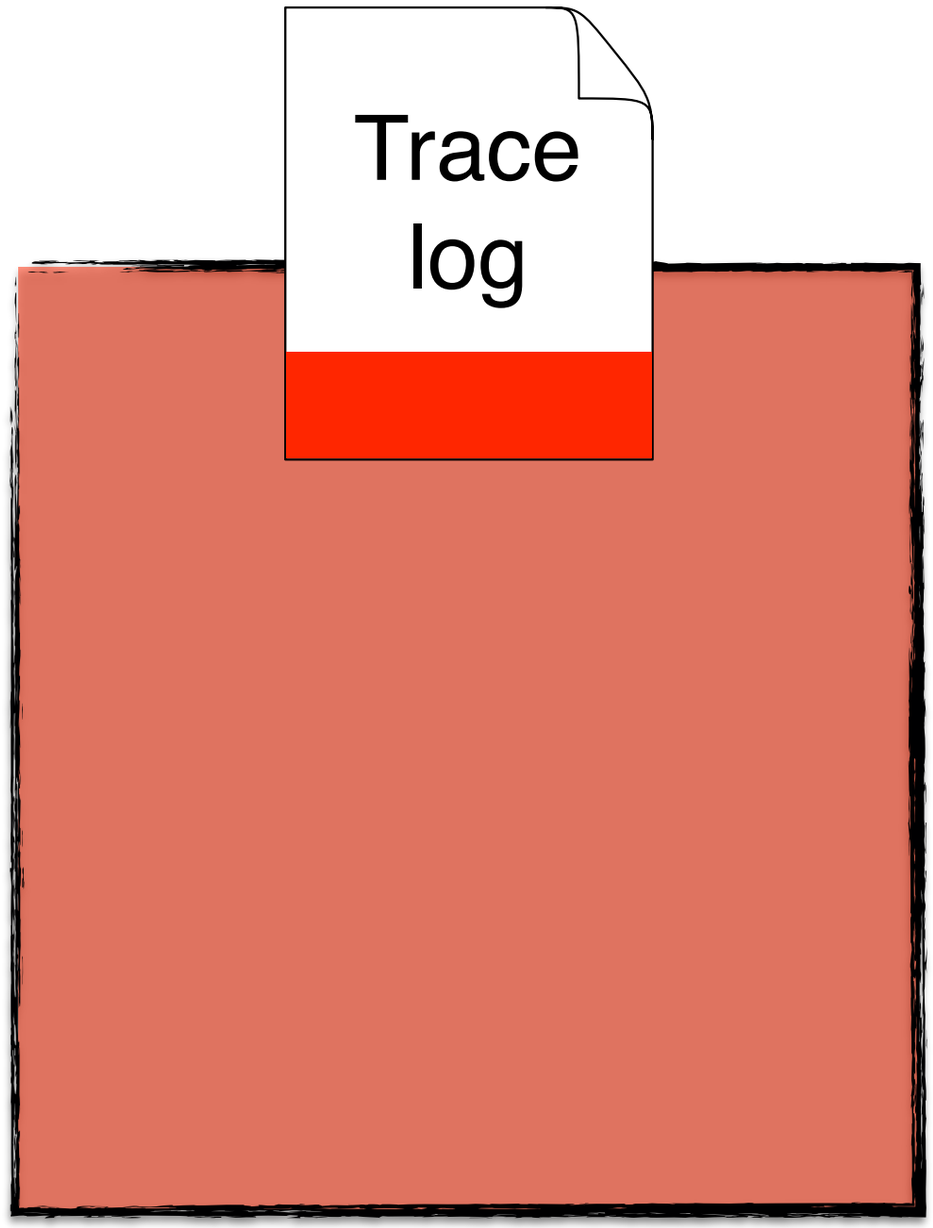
Incompleteness of build specs makes license compliance assessment difficult



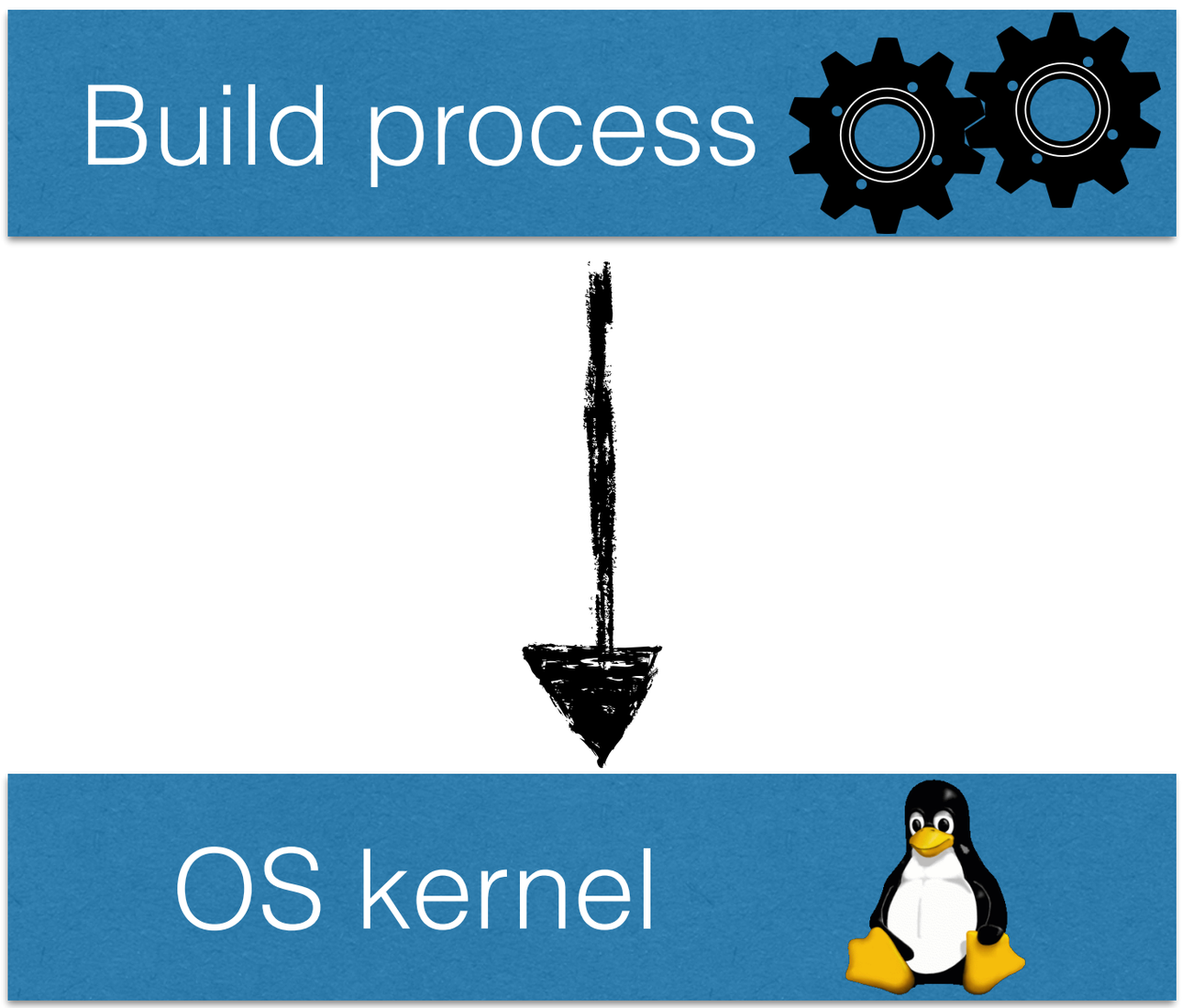
Incompleteness of build specs makes license compliance assessment difficult

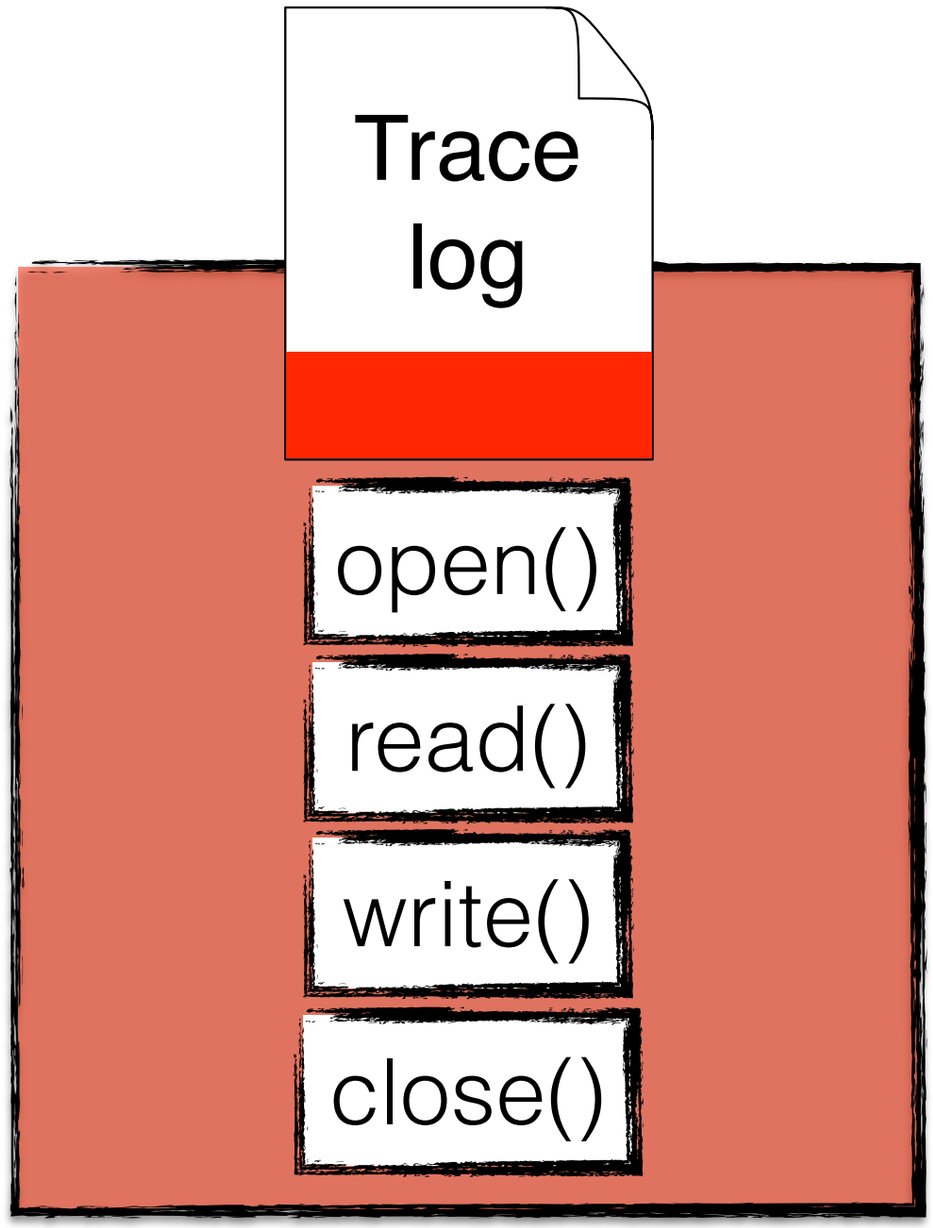


We use system tracing to recover the missing dependencies

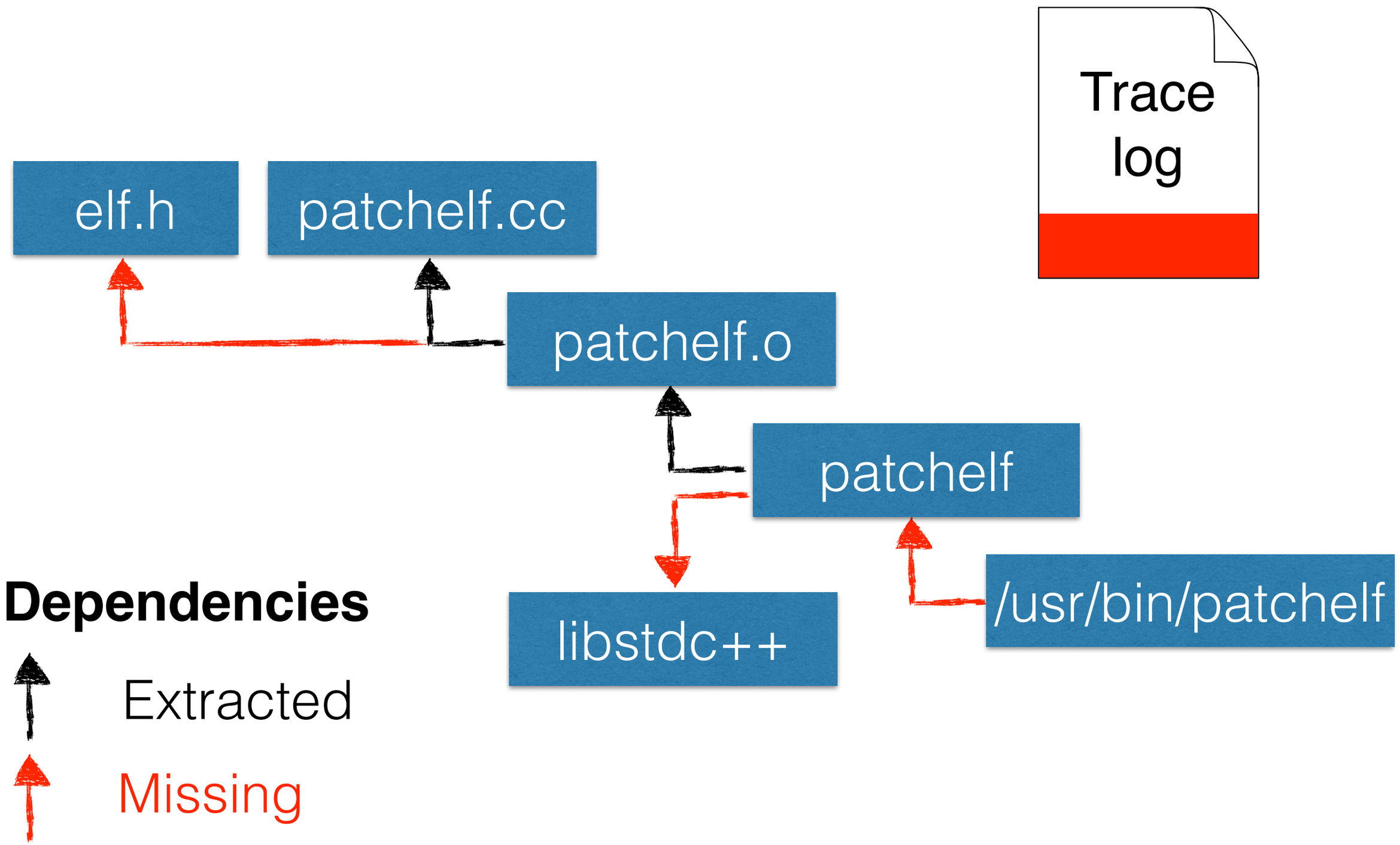


We use system tracing to recover the missing dependencies

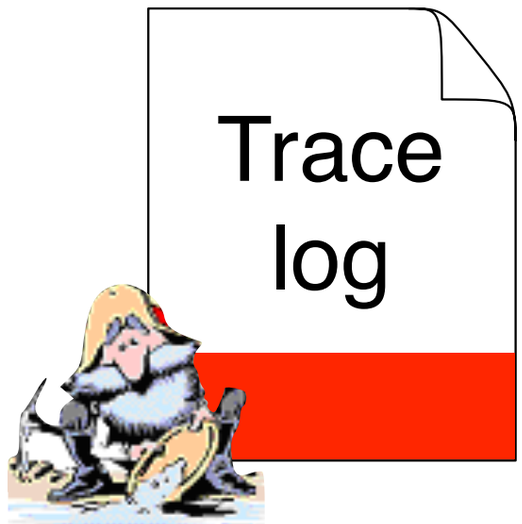




We mine build traces to construct a concrete build dependency graph



We mine build traces to construct a concrete build dependency graph



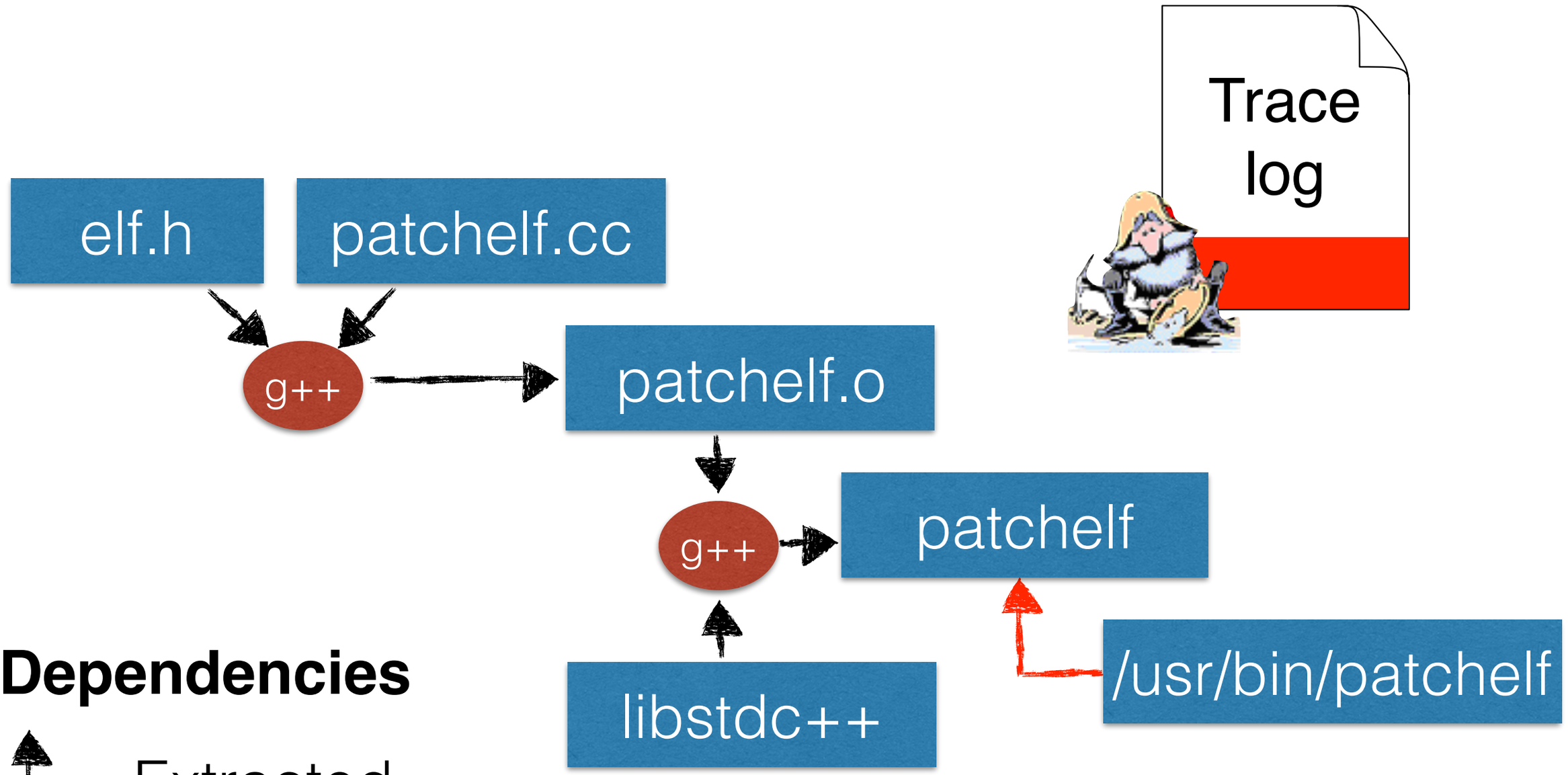
Dependencies

Extracted

Missing

We mine build traces to construct a concrete build dependency graph

We mine build traces to construct a concrete build dependency graph



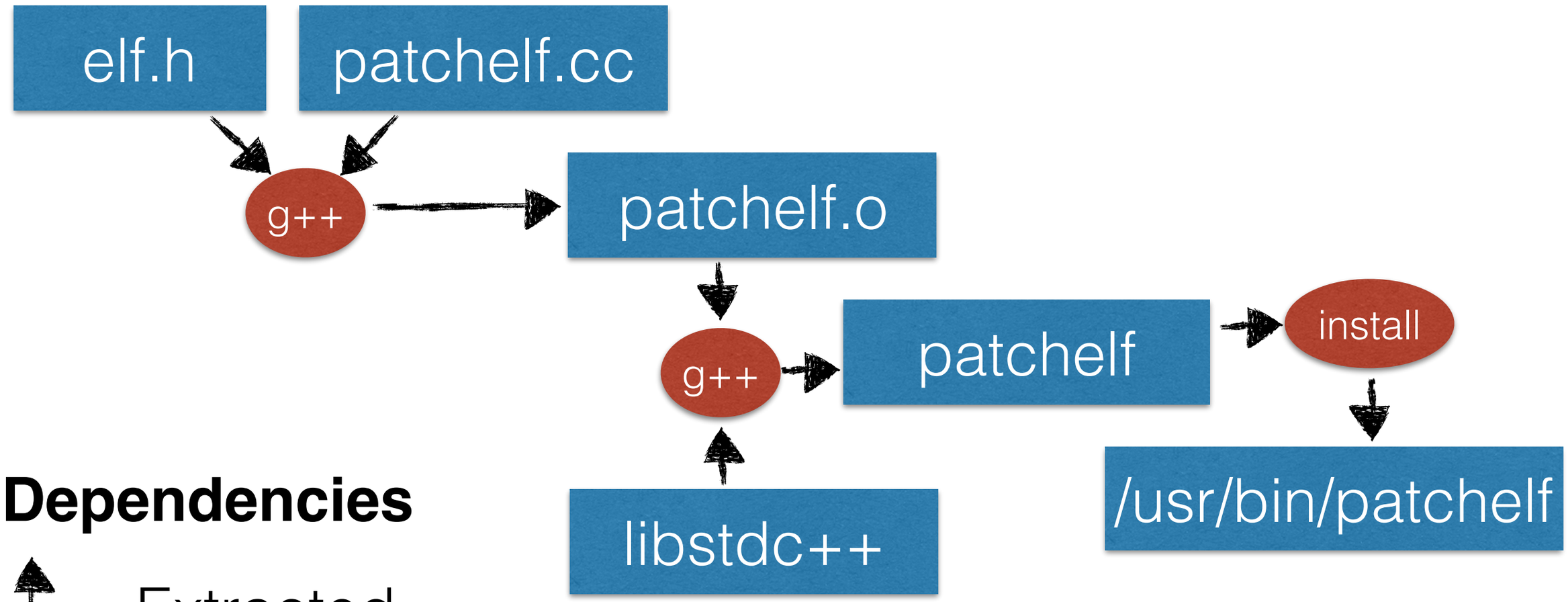
We mine build traces to construct a concrete build dependency graph

Dependencies

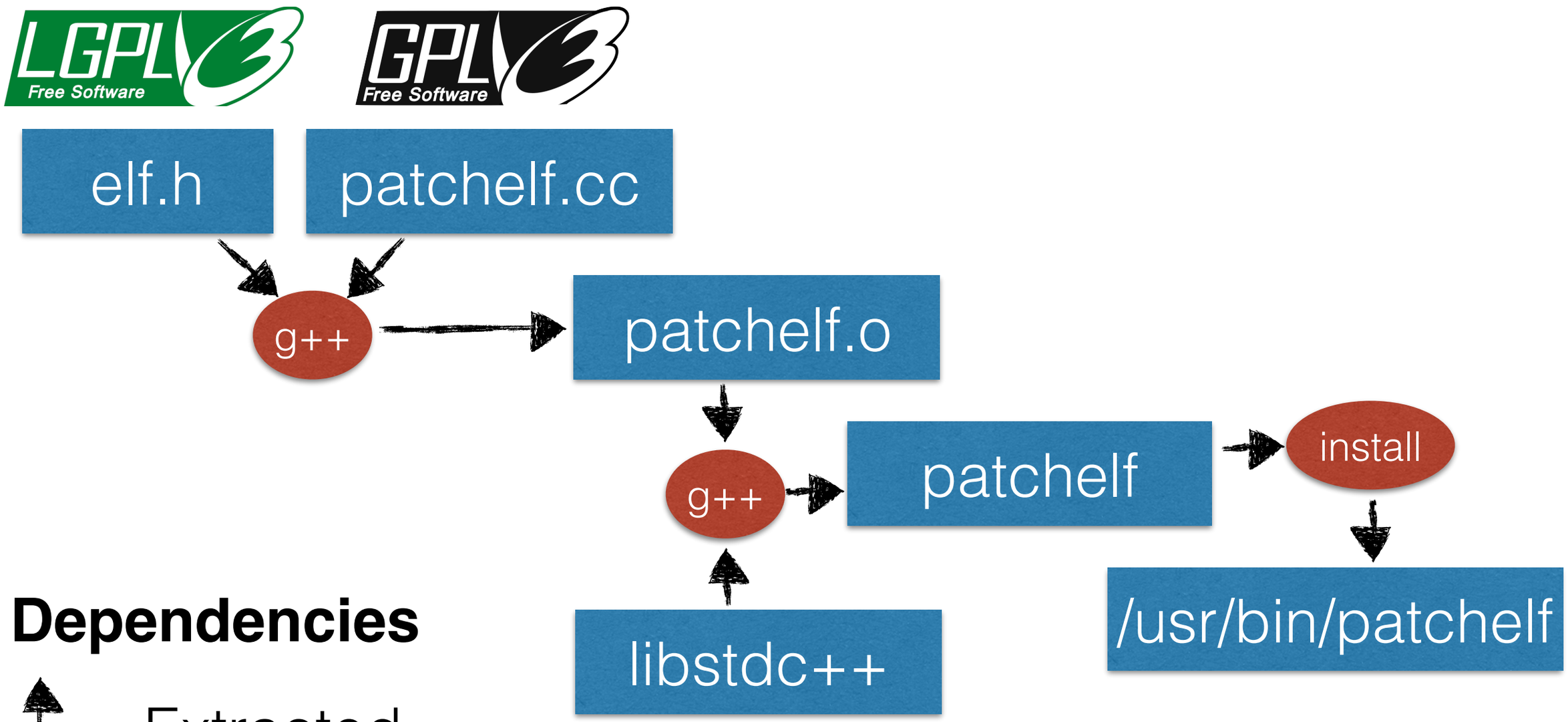
Extracted

Missing

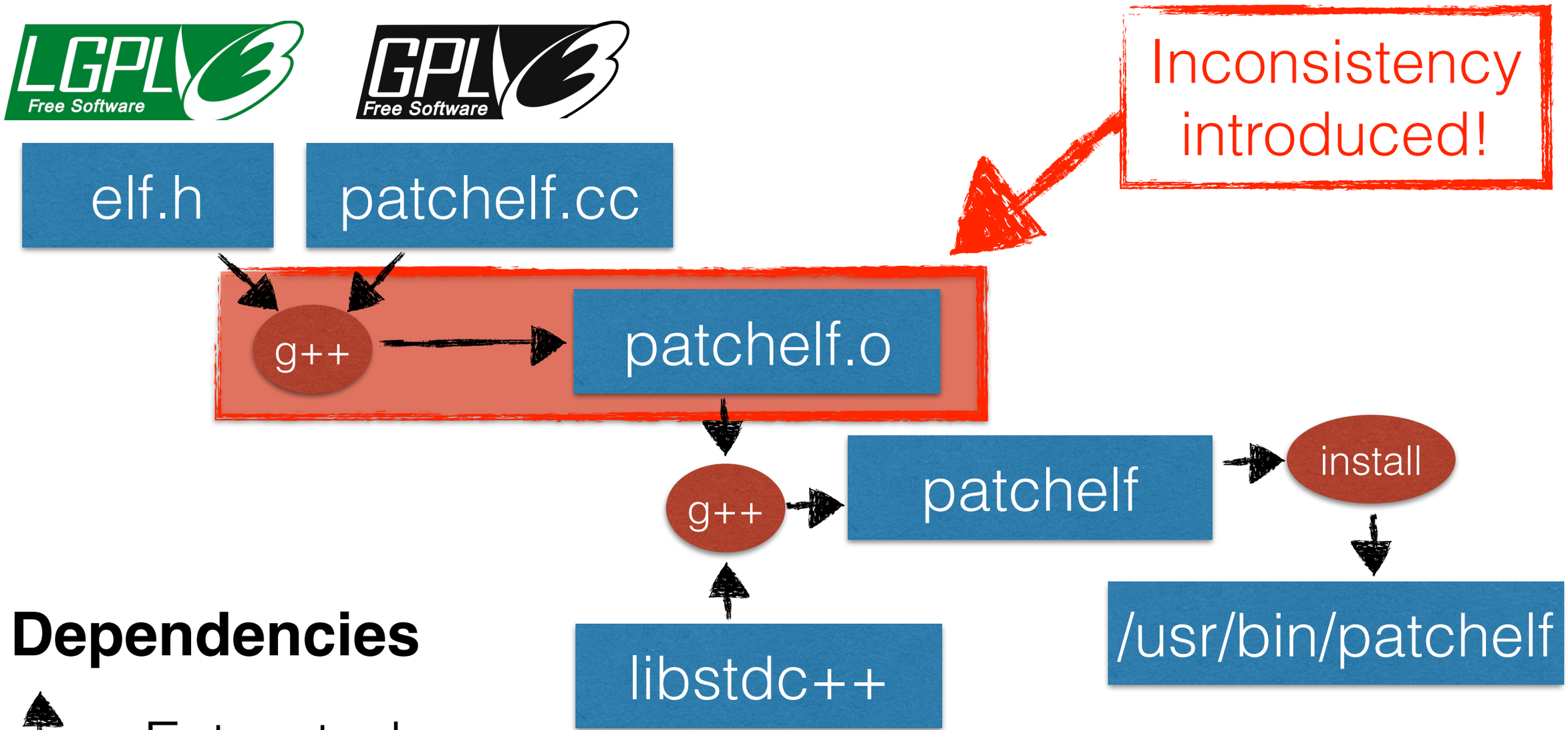
Annotate build graph nodes with license information using Ninka



Annotate build graph nodes with license information using Ninka



Annotate build graph nodes with license information using Ninka



Empirical study

Empirical study

(RQ1)
Accuracy

Empirical study

(RQ1)
Accuracy

(RQ2)
Practicality

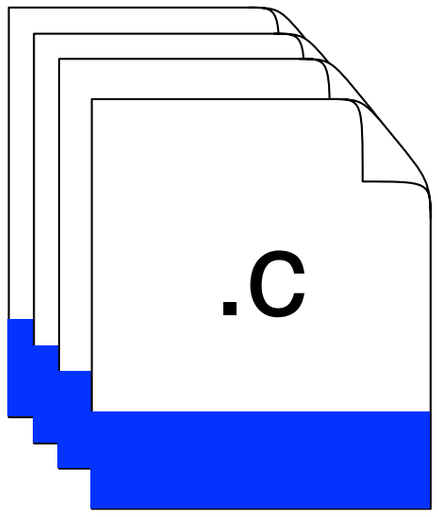
Empirical study

(RQ1) Accuracy

(RQ2) Practicality

Measuring the accuracy of our CBDG approach

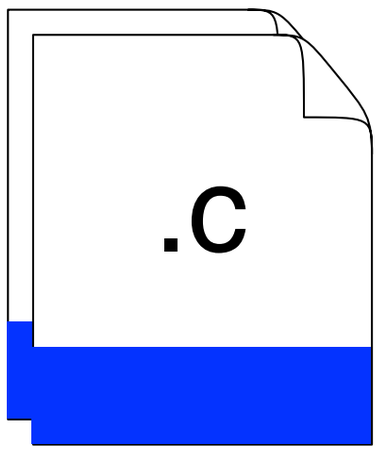
Included



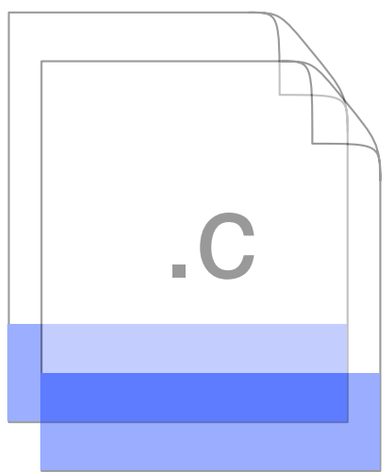
Excluded

Measuring the accuracy of our CBDG approach

Included



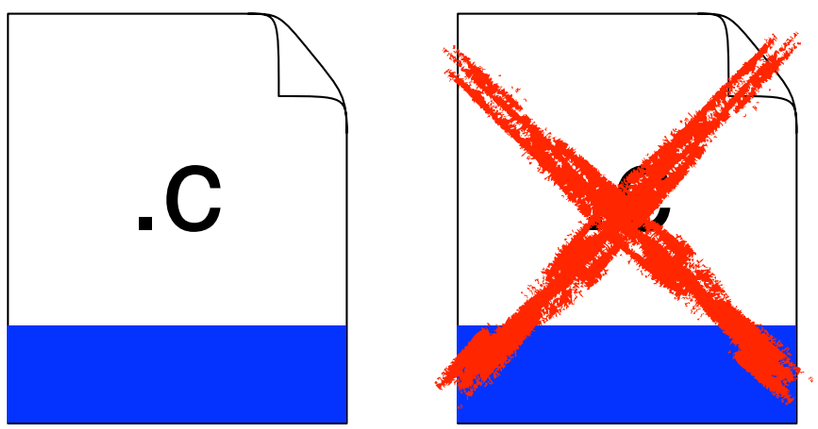
Excluded



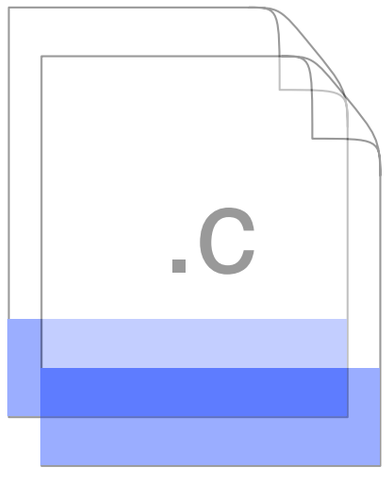
Measuring the accuracy of our CBDG approach

Delete

Included



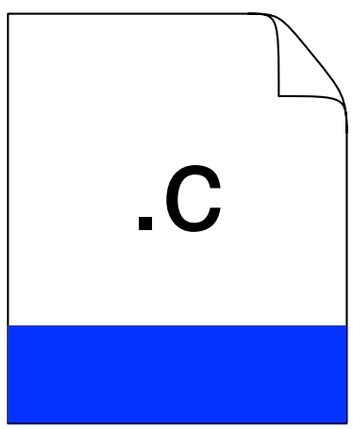
Excluded

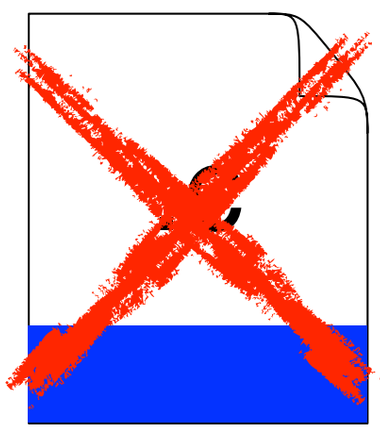


Measuring the accuracy of our CBDG approach

Delete

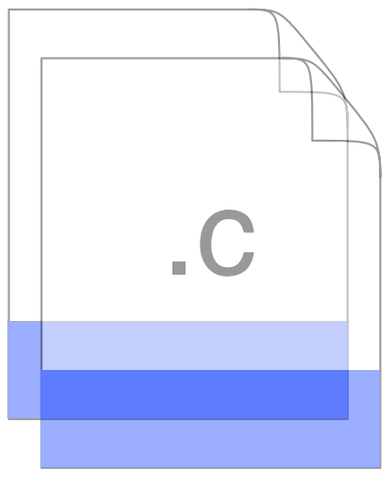
Included





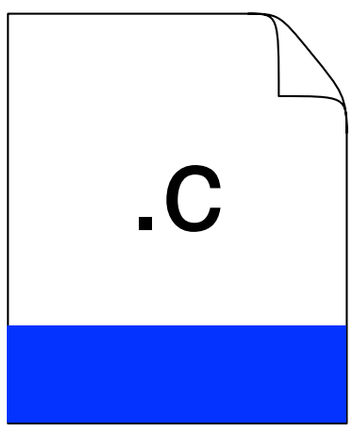
Execute build

Excluded

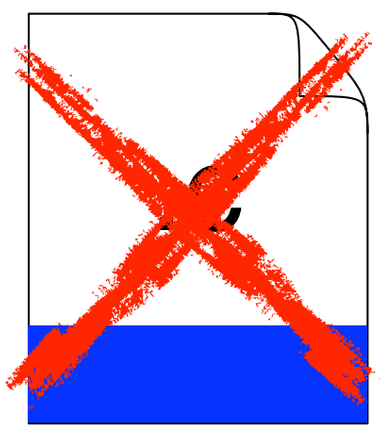


Measuring the accuracy of our CBDG approach

Included



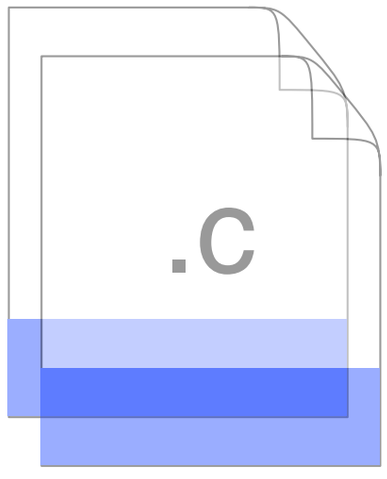
Delete



Execute build

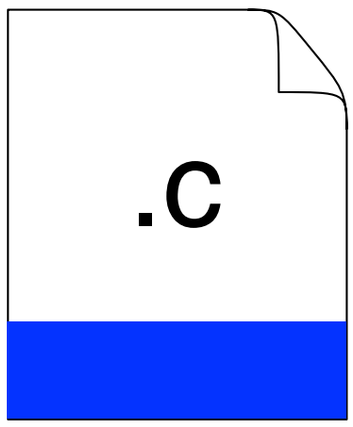
Broken means true positive

Excluded

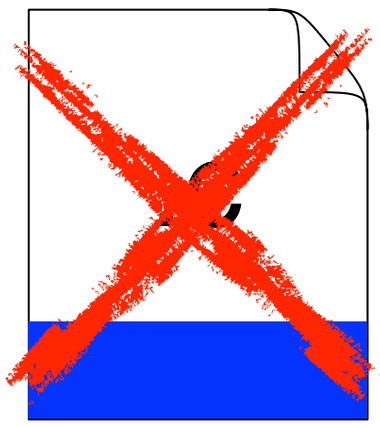


Measuring the accuracy of our CBDG approach

Included



Delete

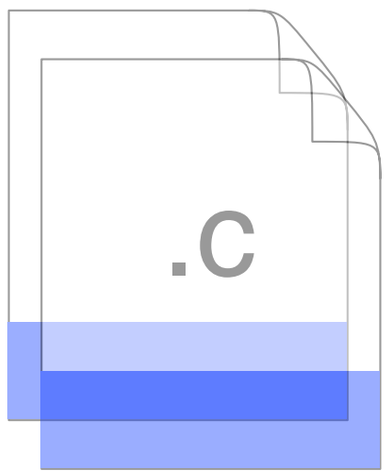


Execute build

Broken means true positive

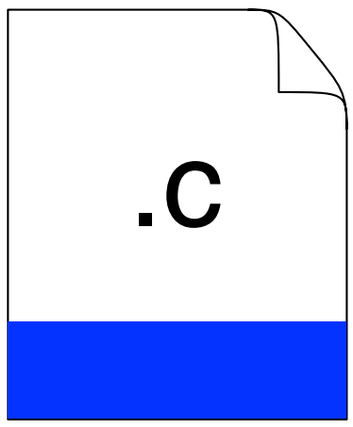
Clean means false positive

Excluded

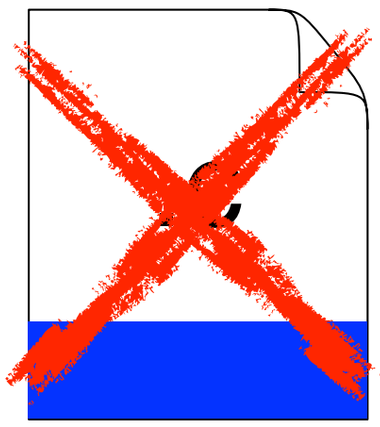


Measuring the accuracy of our CBDG approach

Included



Delete

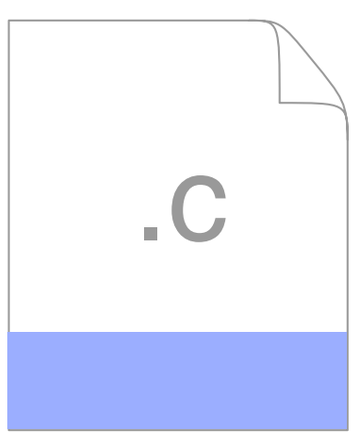


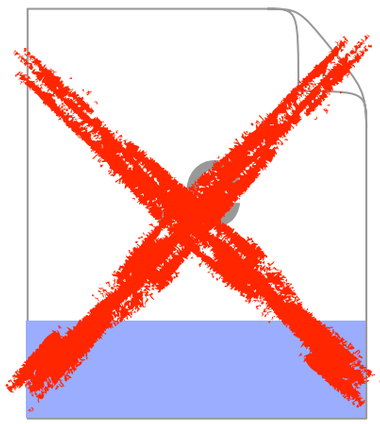
Execute build

Broken means true positive

Clean means false positive

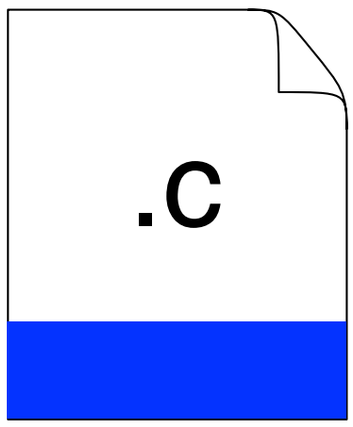
Excluded



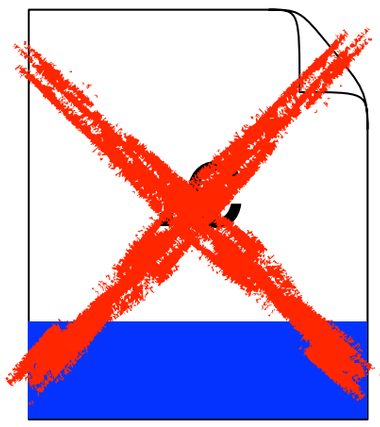


Measuring the accuracy of our CBDG approach

Included



Delete

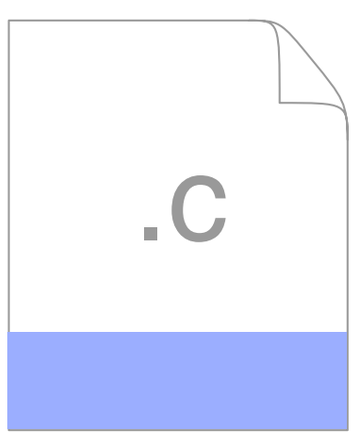


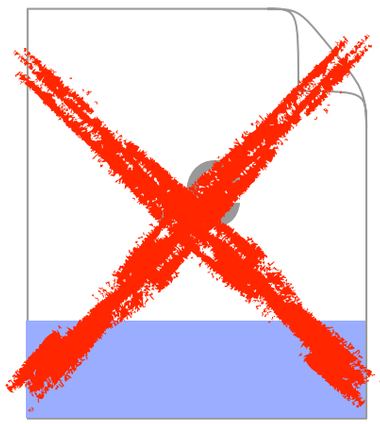
Execute build

Broken means true positive

Clean means false positive

Excluded

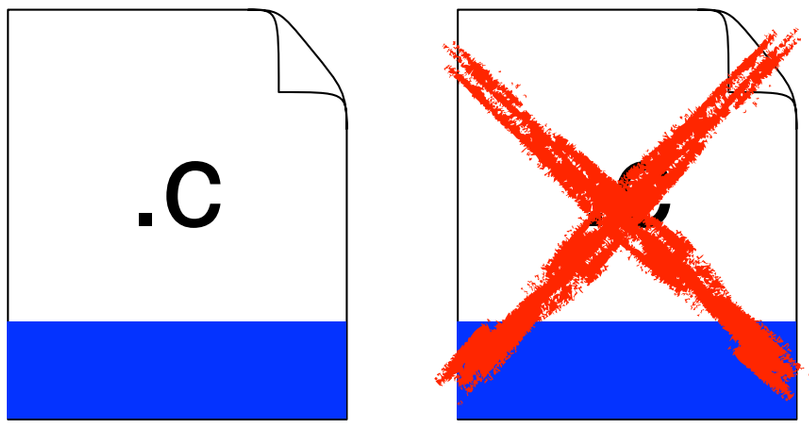




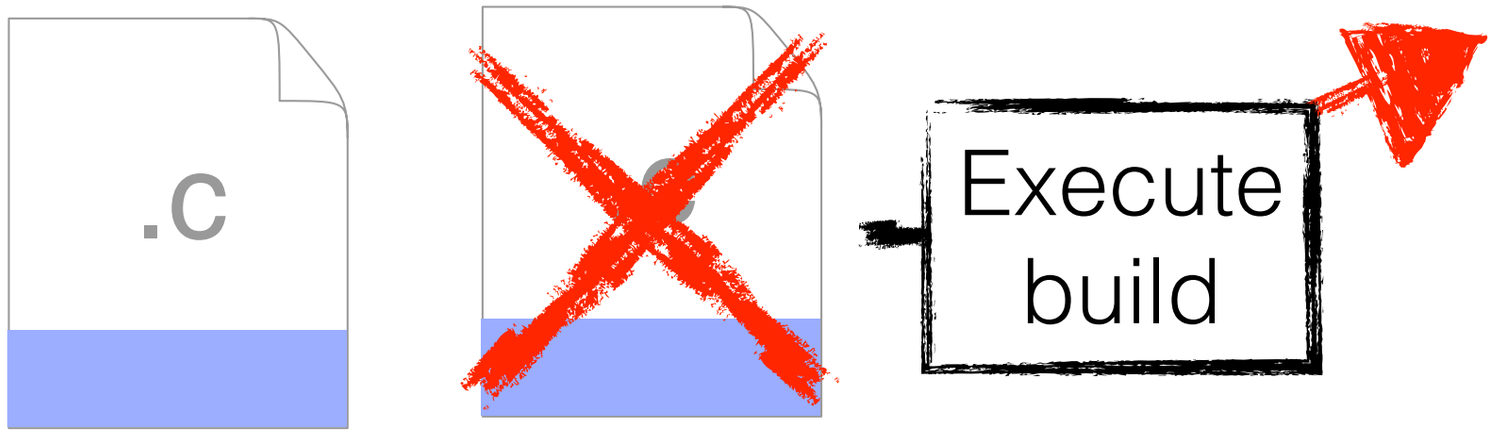
Execute build

Measuring the accuracy of our CBDG approach

Included

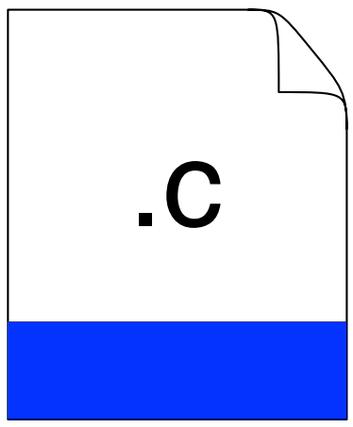


Excluded

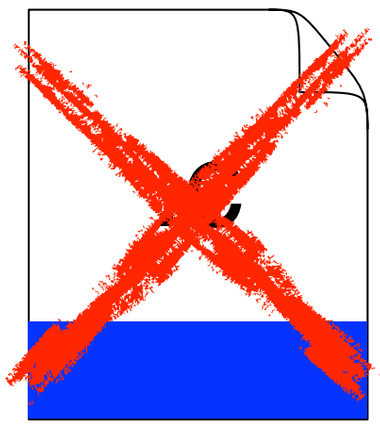


Measuring the accuracy of our CBDG approach

Included



Delete

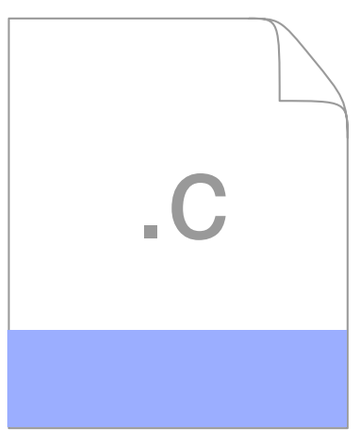


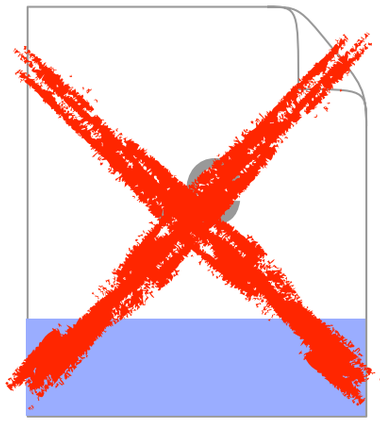
Execute build

Broken means true positive

Clean means false positive

Excluded





Execute build

Broken means false negative

Clean means true negative

Our approach accurately selects the files that impact system deliverables

	Aterm	Opkg	Bash	CUPS	Xalan	OpenSSL	FFmpeg
Technology	Make	Make	Make	Make	Ant	Make	Make
Precision	**100%**	97%	88%	**100%**	99%	99%	99%
Recall	98%	99%	**100%**	99%	**100%**	**100%**	**100%**

But there are cases where our approach makes mistakes

	Aterm	Opkg	Bash	CUPS	Xalan	OpenSSL	FFmpeg
Technology	Make	Make	Make	Make	Ant	Make	Make
Precision	100%	**97%**	**88%**	100%	**99%**	**99%**	**99%**
Recall	98%	99%	100%	99%	100%	100%	100%

And there are cases when our approach misses files that impact deliverables

	Aterm	Opkg	Bash	CUPS	Xalan	OpenSSL	FFmpeg
Technology	Make	Make	Make	Make	Ant	Make	Make
Precision	100%	97%	88%	100%	99%	99%	99%
Recall	**98%**	**99%**	100%	**99%**	100%	100%	100%

Empirical study

(RQ1)
Accuracy

Precision:
88%-100%
Recall:
98%-100%

(RQ2)
Practicality

Empirical study

(RQ1)
Accuracy

Precision:
88%-100%
Recall:
98%-100%

(RQ2)
Practicality

Bugs filed using our approach on multi-licensed packages

FFmpeg

+

License was updated within 3 days

Bugs filed using our approach on multi-licensed packages

FFmpeg

LGPL Free Software

+

GPL Free Software

License was updated within 3 days

CUPS

+

UNIX PRINTING SYSTEM

Offending files were removed within 2 days

Empirical study

(RQ1)
Accuracy

Precision:
88%-100%
Recall:
98%-100%

(RQ2)
Practicality

Prompted quick code changes in two systems

Failure to comply can open an organization up to litigation

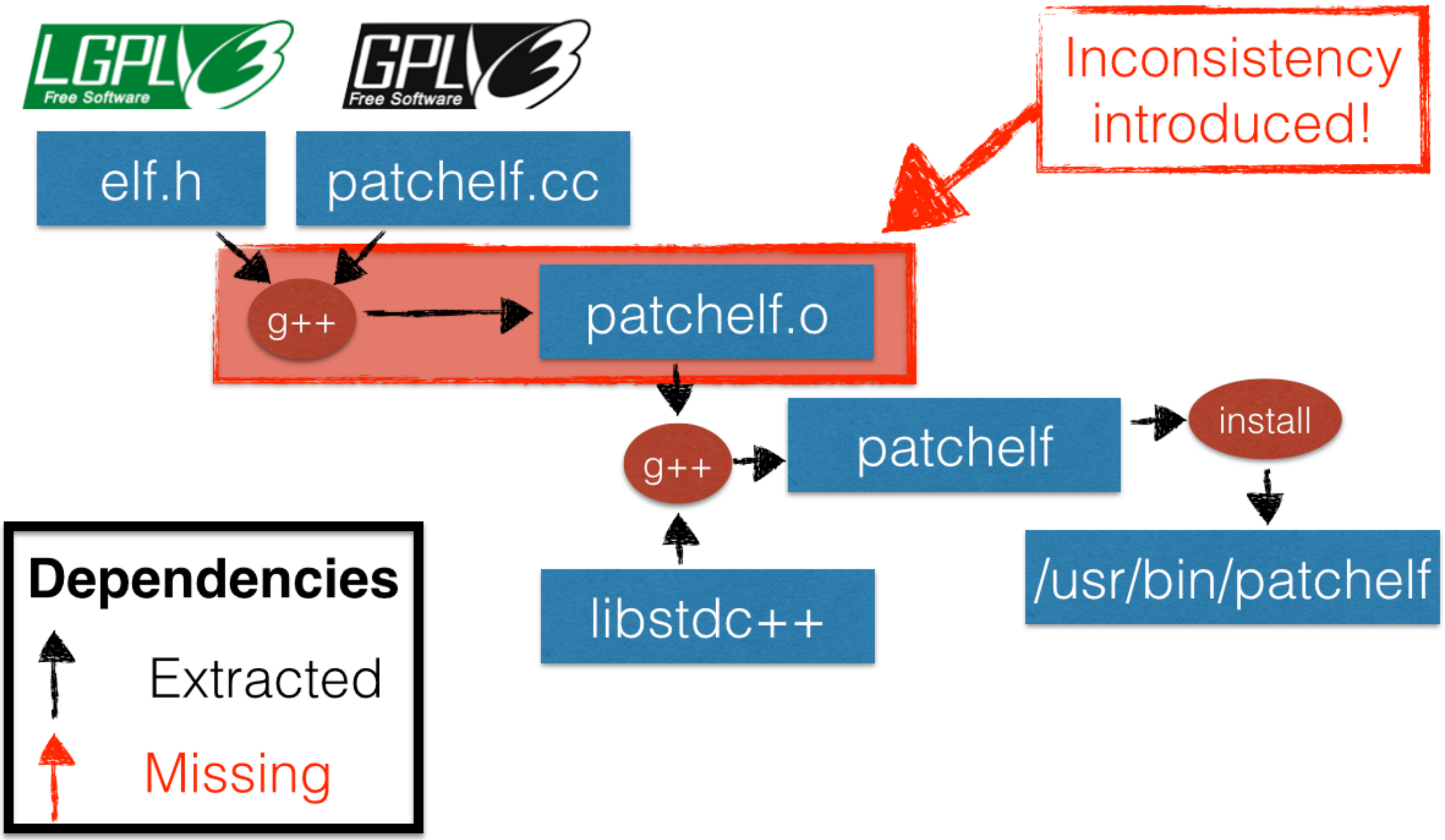
Reuse puts legal constraints on how client systems can be distributed

External component

Implies

Client system

Annotate build graph nodes with license information using Ninka



Empirical study

(RQ1) Accuracy

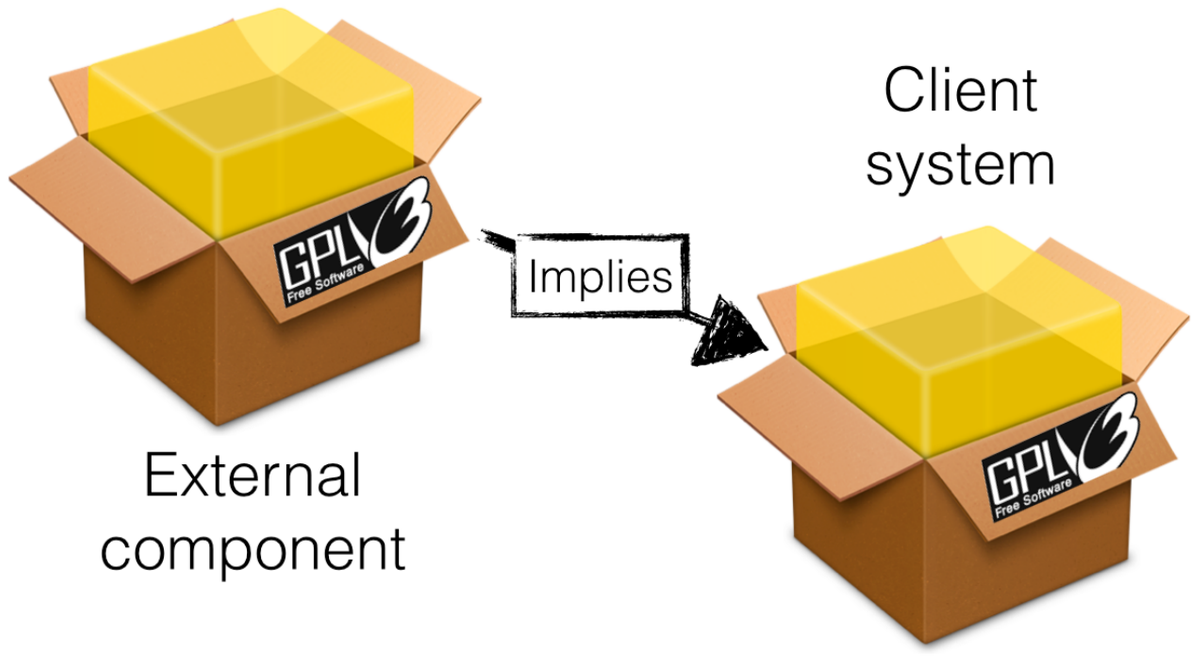
Precision:
0.88-1.0
Recall:
0.98-1.0

(RQ2) Practicality

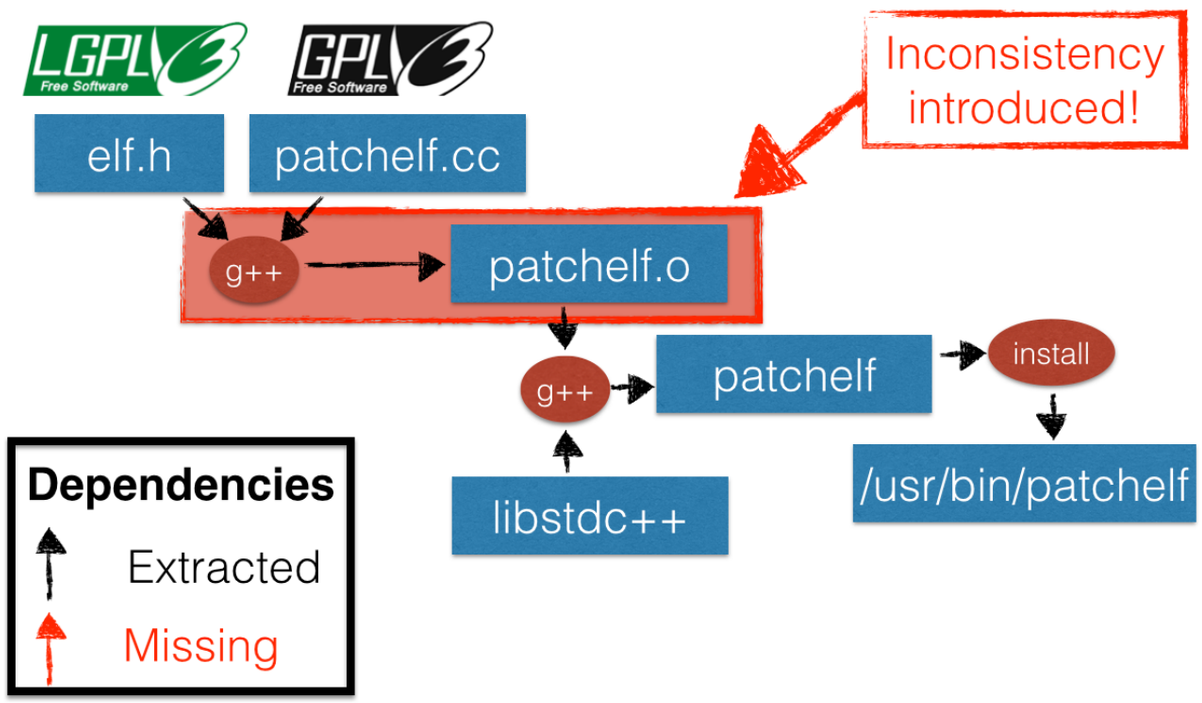
Prompted quick code changes in two systems

Failure to comply can open an organization up to litigation

Reuse puts legal constraints on how client systems can be distributed



Annotate build graph nodes with license information using Ninka



Empirical study

(RQ1)
Accuracy

Precision:
0.88-1.0
Recall:
0.98-1.0

(RQ2)
Practicality

Prompted quick code changes in two systems