GNU Mes – Scheme-only bootstrap

janneke@gnu.org

FOSDEM'20

2020-02-02

Outline

1. Introduction
2. Reproducibility
3. Bootstrappability
4. Thanks

Scheme-only bootstrap: Why?

Why bootstrapping is important to you.
or
Why bootstrapping is something you wish to ignore.

Scheme-only bootstrap: GNU Mes

GNU Mes

- A Scheme interpreter written in ~5,000LOC of simple C.
- A C compiler written in Scheme.
- Built on LISP: eval/apply, the Maxwell Equations of Software.

Auditable Elegance

```
(define (apply fn x a)
  (cond
   ((atom fn)
    (cond
     ((eq fn CAR)  (caar x))
     ((eq fn CDR)  (cdar x))
     ((eq fn CONS) (cons (car x) (cadr x)))
     ((eq fn ATOM) (atom (car x)))
     ((eq fn EQ)   (eq (car x) (cadr x)))
     (#t           (apply (eval fn a) x a))))
   ((eq (car fn) LAMBDA)
                 (eval (caddr fn) (pairlis (cadr fn) x a)))
   ((eq (car fn) LABEL)
                 (apply (caddr fn) x
                        (cons (cons (cadr fn) (caddr fn)) a)))))

(define (eval e a)
  (cond
   ((atom e) (cdr (assoc e a)))
   ((atom (car e))
    (cond ((eq (car e) QUOTE) (cadr e))
          ((eq (car e) COND)  (evcon (cdr e) a))
          (#t                 (apply (car e) (evlis (cdr e) a) a))))
   (#t       (apply (car e) (evlis (cdr e) a) a))))
```

`eval` and `apply` are mutual recursing functions that—using a few helper functions—describe the universe of computing.

Long path: Best Practice

- 500+ MB: no bootstrap

Guix

Pronounced *Geeks*

Reduce binary seeds to bare minimum

These big chunks of binary code are practically non-auditable which breaks the source to binary transparency that we get in the rest of the package dependency graph.

Every unauditable binary leaves us vulnerable to compiler backdoors as described by Ken Thompson in the 1984 paper Reflections on Trusting Trust.

Thus, our goal is to reduce the set of bootstrap binaries to the bare minimum. – Ludovic Courtès (GNU Guix documentation, December 2017)

Ken Thompson

TURING AWARD LECTURE

Reflections on Trusting Trust

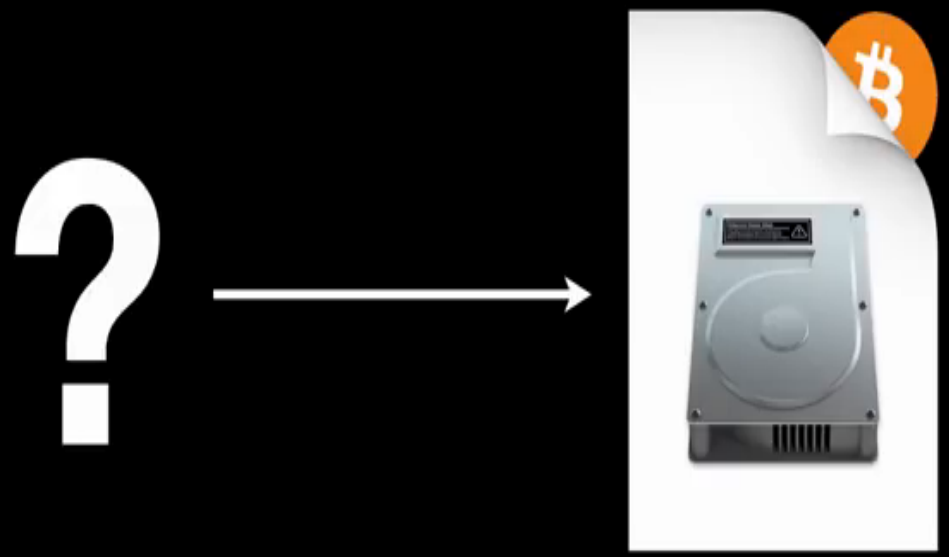
To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.

Long path: Ignoring the Problem

- 500+ MB: no bootstrap

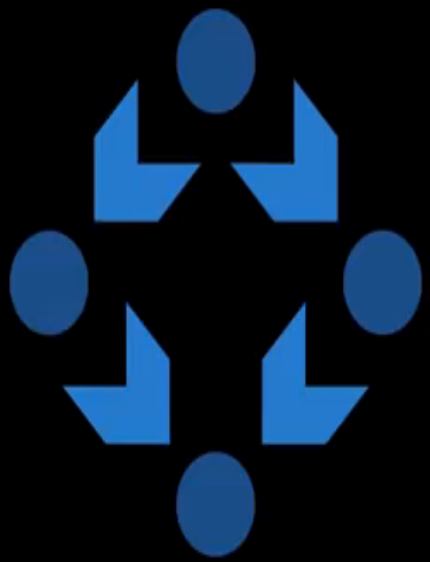
Long path: GNU Guix System v1.0

- 500+ MB: no bootstrap
- 252 MB: GNU Guix System v1.0



Bitcoin Build System Security

Carl Dong, Chaincode Labs



Reproducible Builds

What is a Bootstrap?

Impossible task: pull yourself up on your boot straps

Software: to create your first: kernel, shell, C compiler, ...

source + ?? =

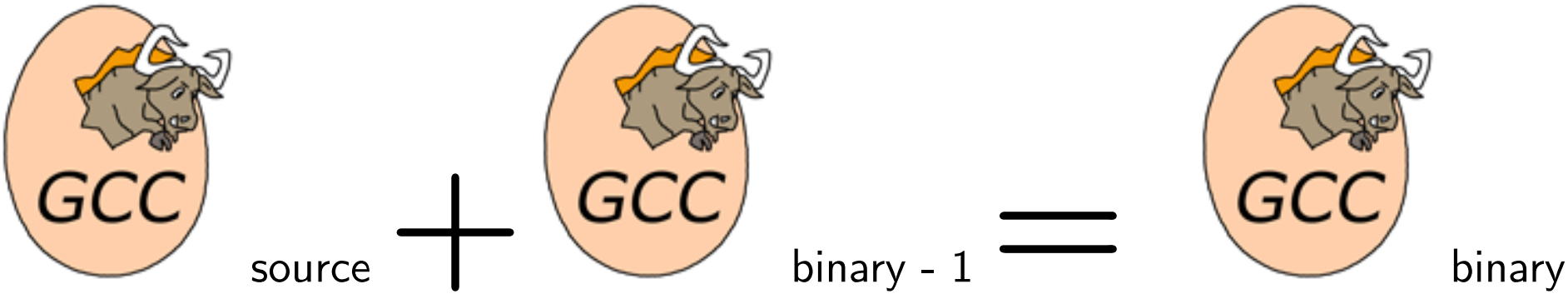
binary

How to Bootstrap: An Old Recipe...

Recipe for yoghurt: Add yoghurt to milk – Anonymous

How to Bootstrap: Create your second GCC

Traditional recipe: like yoghurt



. . . and done!

We're Reproducible!

We're Reproducible!

We're Reproducibly Malicious

Reproducibility **is not enough**

Reproducibility
Clean source code

is not enough

Guix

Pronounced *Geeks*

Long path: Reduced Binary Seed bootstrap

- 500+ MB: no bootstrap
- 252 MB: GNU Guix System v1.0
- 145 MB: Reduced Binary Seed
  - master branch
  - ~~GCC~~, ~~GLIBC~~, ~~Binutils~~
  - + MesCC-Tools, + Mes

NixOS


gcc0
glibc0
gcc-core
mes-headers
bootstrap-linux-libre-headers
binutils0
diffutils
make0
tcc
tcc0
mes
bootstrap-mescc-tools
bootstrap-mes
bootstrap-binaries
bash
bzip2
coreutils
gawk
grep
gzip
patch
sed
tar
xz

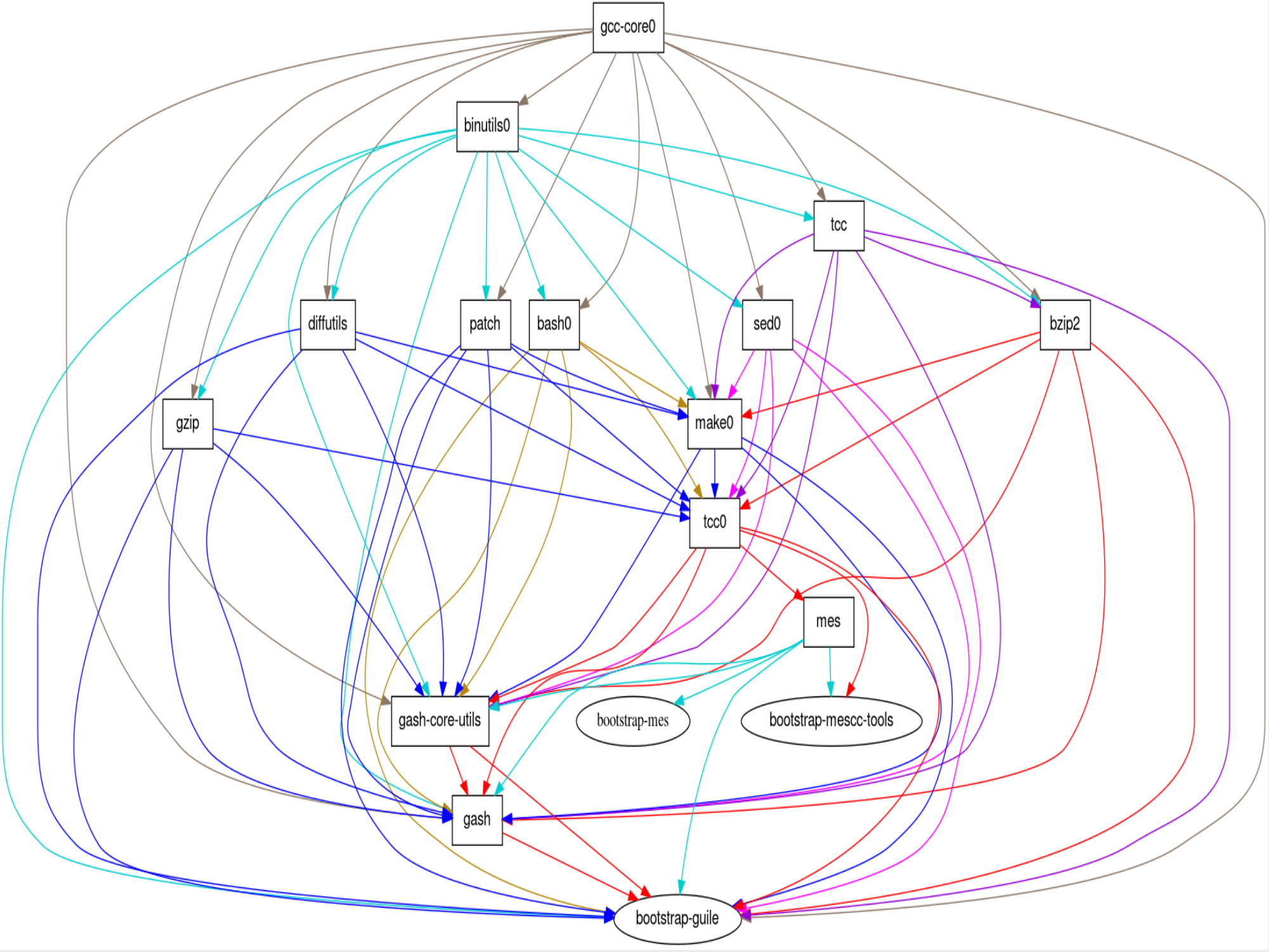
NLnet Foundation

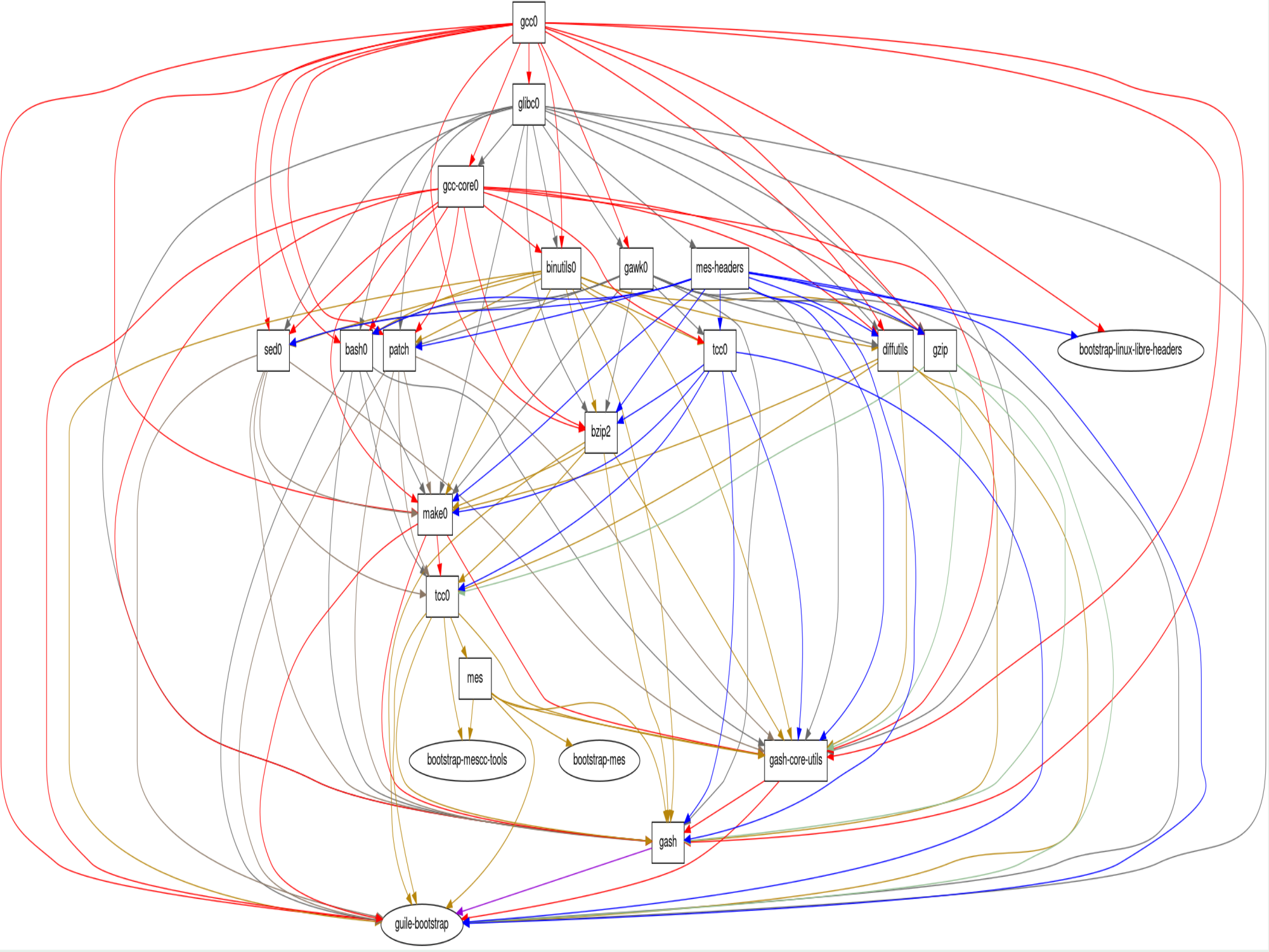
Long path: Scheme-only bootstrap

- 500+ MB: no bootstrap
- 252 MB: GNU Guix System v1.0
- 145 MB: Reduced Binary Seed
  - master branch
  - ~~GCC~~, ~~GLIBC~~, ~~Binutils~~
  - + MesCC-Tools, + Mes
- 57 MB: Scheme-only
  - wip-bootstrap branch
  - ~~Awk~~, ~~Bash~~, ~~Bzip2~~, ~~GNU Core Utilities~~, ~~Grep~~, ~~Gzip~~, ~~Make~~, ~~Patch~~, ~~Sed~~, ~~Tar~~, and ~~XZ~~.
  - + Gash (source only!)

Scheme-only bootstrap: Gash Core Utils

awk	cp	gash	mv	sleep	uname
basename	cut	grep	pwd	sort	uniq
bash	diff	gzip	reboot	tar	wc
cat	dirname	head	rm	test	which
chmod	expr	ln	rmdir	touch	
cmp	false	ls	sed	tr	
compress	find	mkdir	sh	true	


gcc-core0
binutils0
tcc
diffutils
patch
bash0
sed0
bzip2
gzip
make0
tcc0
mes
gash-core-utils
bootstrap-mes
bootstrap-mescc-tools
gash
bootstrap-guile


gcc0
glibc0
gcc-core0
binutils0
gawk0
mes-headers
sed0
bash0
patch
tcc0
diffutils
gzip
bootstrap-linux-libre-headers
bzip2
make0
tcc0
mes
bootstrap-mescc-tools
bootstrap-mes
gash-core-utils
gash
guile-bootstrap

Cross distro reproducibility

The sha256sum for bin/mes-mescc on x86 shall be

```
722790ed261954eb53cf2cd2906c89c7589ef72b66171bbe2a9dce0f0af20232 v0.22
9e0bcb1633c58e7bc415f6ea27cee7951d6b0658e13cdc147e992b31a14625fb v0.21
```

only differing in the version number string.

For v0.21 this has been verified on Guix System, Debian GNU/Linux and NixOS.

The holy grail

The holy grail of bootstrappability will be connecting mes to hex0.
– Carl Dong, Chaincode Labs

Full Source Bootstrap

Long path: Full Source Bootstrap

- 500+ MB: no bootstrap

- 252 MB: GNU Guix System v1.0

- 145 MB: Reduced Binary Seed
  - master branch
  - ~~GCC~~, ~~GLIBC~~, ~~Binutils~~
  - + MesCC-Tools, + Mes

- 57 MB: Scheme-only
  - wip-bootstrap branch
  - ~~Awk~~, ~~Bash~~, ~~Bzip2~~, ~~GNU Core Utilities~~, ~~Grep~~, ~~Gzip~~, ~~Make~~, ~~Patch~~, ~~Sed~~, ~~Tar~~, and ~~XZ~~.
  - + Gash (source only!)

- 357 bytes: Full Source
  - ~~MesCC-Tools~~, ~~Mes~~
  - + Stage0: 357 bytes (x86)

Trusted Computing Base

- Source code
- Binary seeds
- Guix System
- Linux

Raising the bar on auditibility

I want code easy to reason about at the heart of this bootstrap, so that everyone will be able to sit down in the morning and be done by lunch time; understanding how every piece of it works. – Jeremiah Orians

Won't your life be boring?

MesCC should optimize for the ease of convincing us of its correctness. – Mark H Weaver

Joy of Source

Vulnerability to a trusting trust attack is a symptom of an unauditable or missing bootstrap story. – janneke

Thanks

- Carl Dong
- Danny Milosavljevic
- David Terry
- Jeremiah Orians
- Ludovic Courtès
- Matt Wette
- Pjotr Prins
- Rutger van Beusekom
- Timothy Sample
- Vagrant Cascadian

Want to join?

You can help

- make Guix run on Mes
- write a bootstrappable syntax-case
- simplify MesCC and target GCC-4.6
- bootstrap NixOS, Debian
- port MesCC to the Hurd, FreeBSD
- spread the message
- retweet `@janneke_gnu janneke@octodon.social`

Connect

- irc freenode.net #bootstrappable #guix
- mail bug-mes@gnu.org guix-devel@gnu.org
- git https://git.savannah.gnu.org/git/mes.git
- web bootstrappable.org