2008.7.5

JNUG 2008 BOF

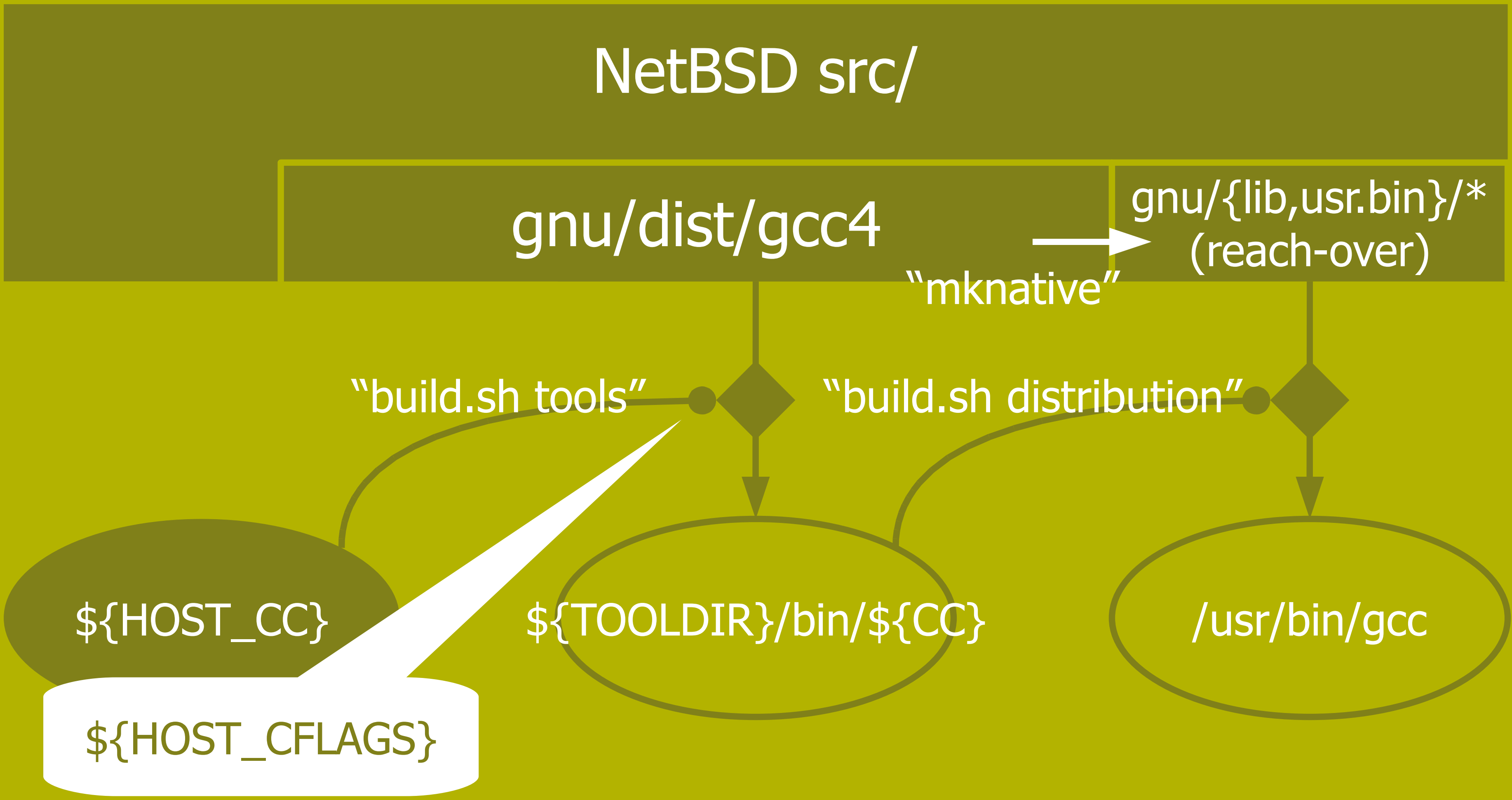
Toolchain and “mknative”

Masao Uebayashi <uebayasi@{gmail.com,NetBSD.org}>

What is mknative?

- Scripts (tools/*/*mknative*)
- To generate reach-overs for binutils + GCC + GDB
- Considering...
  - build, host, target
  - Canadian cross
  - GNU Autoconf, GNU make -> BSD make
  - ...

Compilers - ${HOST_CC}, ${CC}, gcc



Compilers - ${HOST_CC}

- Pre-existing
- Not necessarily GCC
- Compiles ${CC} which compiles NetBSD source tree
- Native compiler for build system

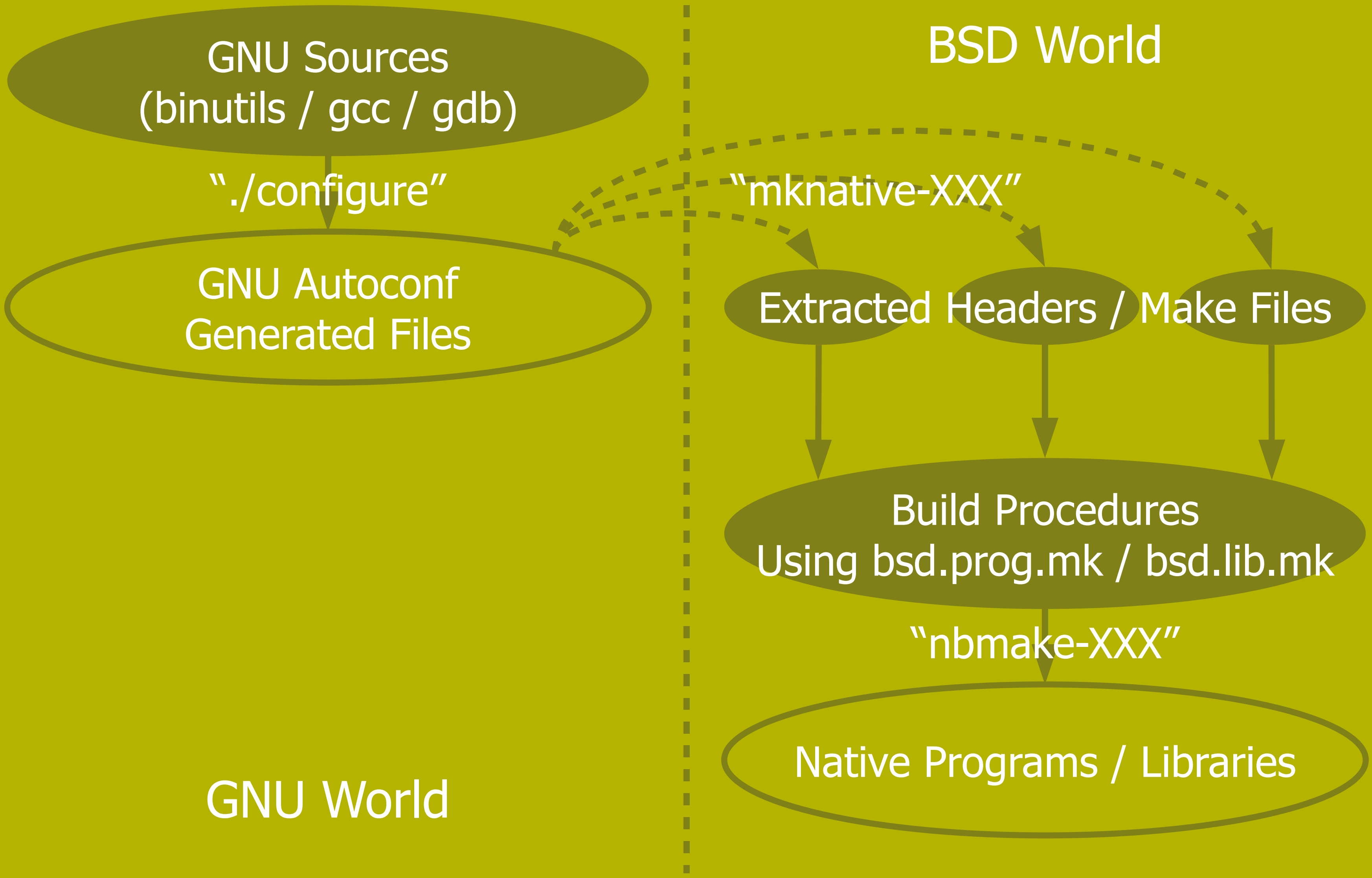
Compilers - ${CC}

- Built from gnu/dist/gcc4
- Installed into ${TOOLDIR}/bin
- Compiles NetBSD source tree
- Cross compiler

Compilers - gnu/usr.bin/gcc4/*

- Built from gnu/dist/gcc4
- Installed into ${DESTDIR}/usr/bin
- Native compiler for target system

Whole Picture



GNU Sources (binutils / gcc / gdb)

"./configure"

GNU Autoconf Generated Files

GNU World

BSD World

"mknative-XXX"

Extracted Headers / Make Files

Build Procedures Using bsd.prog.mk / bsd.lib.mk

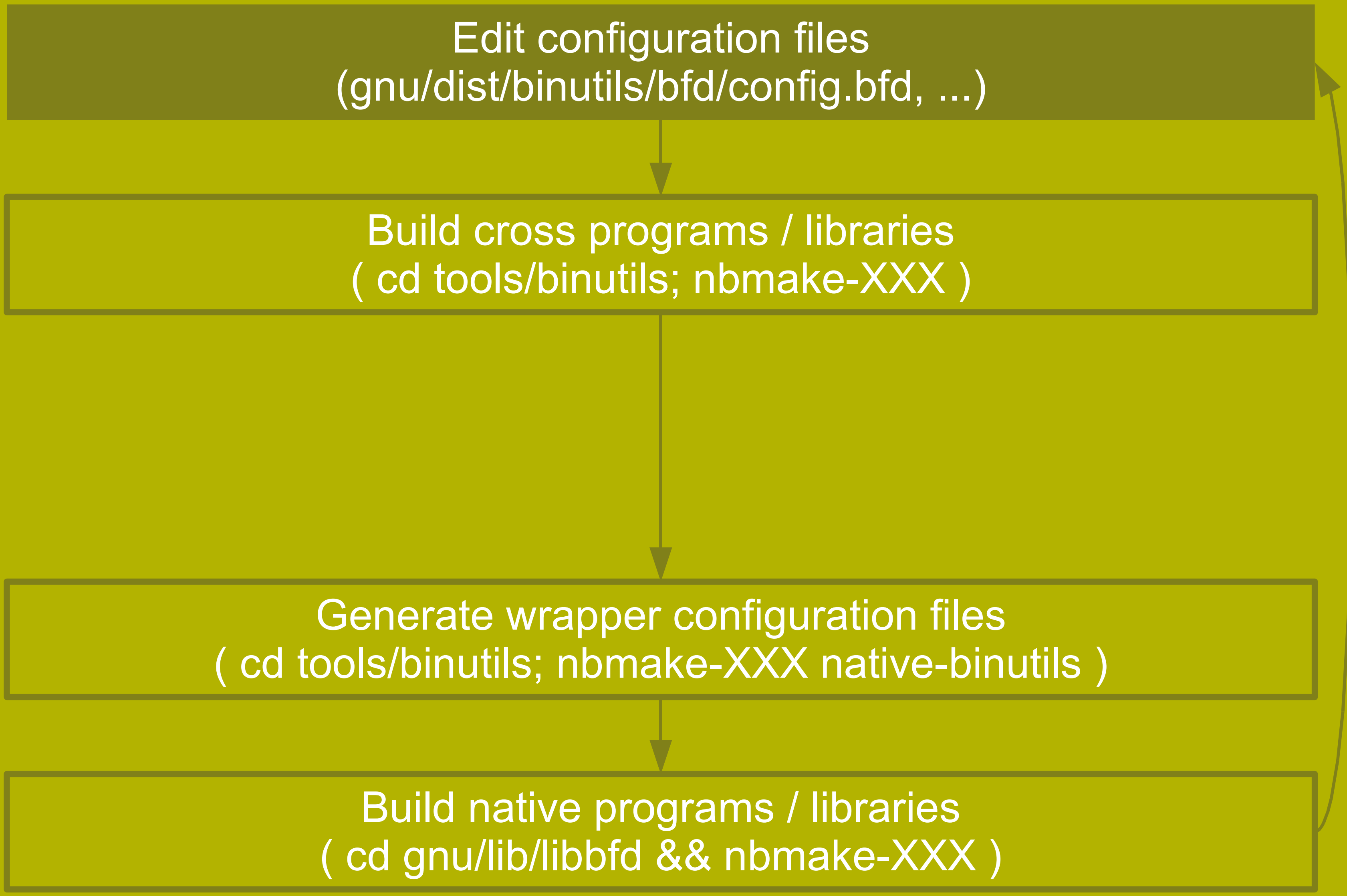
"nbmake-XXX"

Native Programs / Libraries

Toolchain Development

- Change version?
  - No
    - easy
    - edit config files in dist/
    - re-run mknative
  - Yes
    - hard
    - edit config files in dist/
    - !@#$%
    - re-run mknative

Toolchain Development (easy)



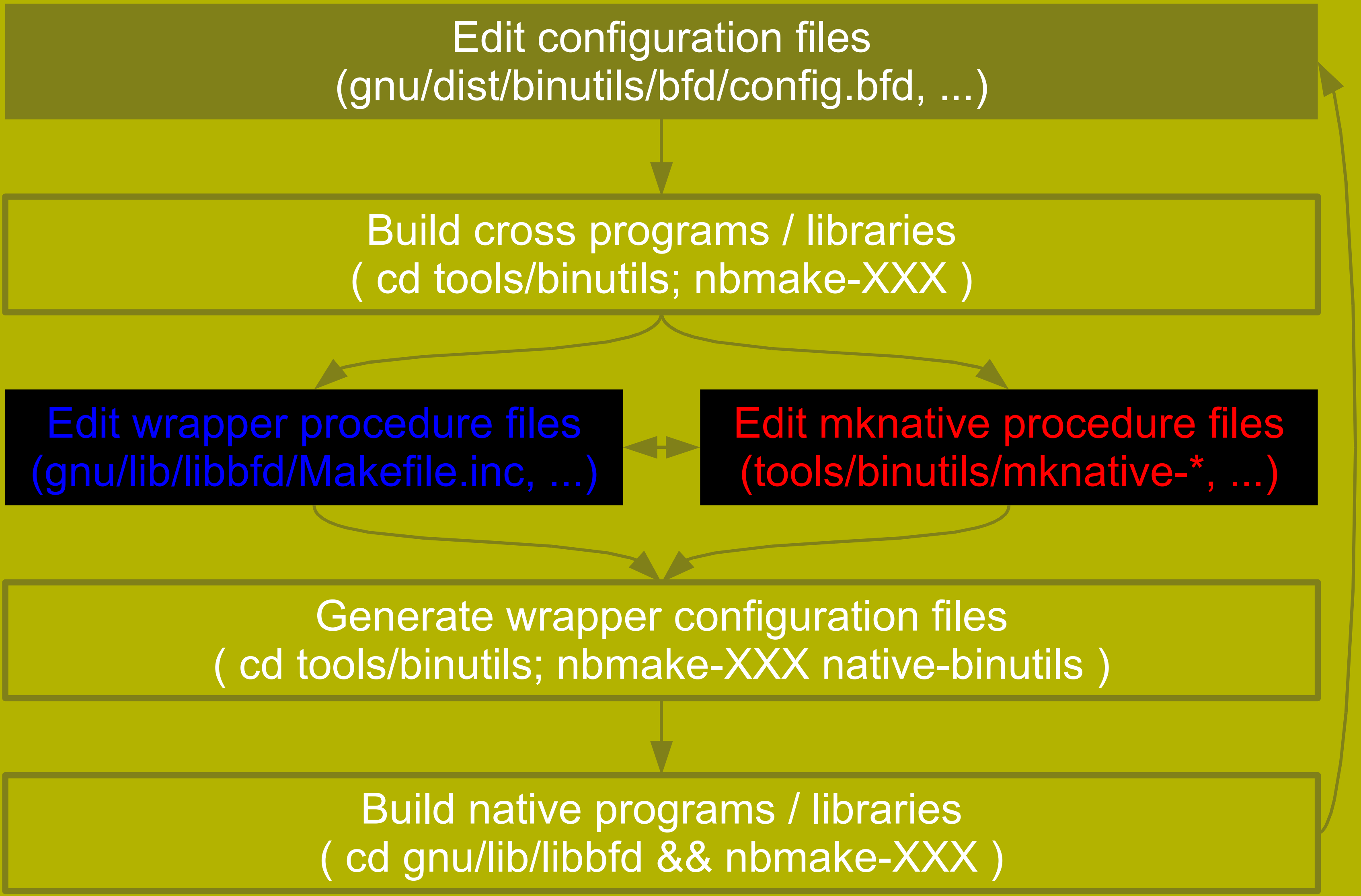
Edit configuration files
(gnu/dist/binutils/bfd/config.bfd, ...)

Build cross programs / libraries
(cd tools/binutils; nbmake-XXX)

Generate wrapper configuration files
(cd tools/binutils; nbmake-XXX native-binutils)

Build native programs / libraries
(cd gnu/lib/libbfd && nbmake-XXX)

Toolchain Development (hard)



Edit configuration files
(gnu/dist/binutils/bfd/config.bfd, ...)

Build cross programs / libraries
(cd tools/binutils; nbmake-XXX)

Edit wrapper procedure files
(gnu/lib/libbfd/Makefile.inc, ...)

Edit mknative procedure files
(tools/binutils/mknative-*, ...)

Generate wrapper configuration files
(cd tools/binutils; nbmake-XXX native-binutils)

Build native programs / libraries
(cd gnu/lib/libbfd && nbmake-XXX)

tools/gcc/mknative.common

- To extract make file values:

```
getvars()
{
   make -f - -f “${XXX}” _x_ <<EOF
_x_:
.for vars in $@
@echo G_\${var}=\${\${var}:Q} | sed ...
.endfor
.include "$_TMPDIR/$_mf"
EOF
}
```

- To extract shell file values:

```
sh -c '. hoge.sh >/dev/null; echo $fuga'
```

- grep, sed, ...

tools/binutils/mknative-binutils

- To write *.mk / *.h files...

```
get_libbfd() {
   mkdir -p $_TOP/gnu/lib/libbfd/arch/$MACHINE_ARCH

   {
      getvars bfd/Makefile \
         libbfd_la_DEPENDENCIES libbfd_la_OBJECTS DEFS \
         INCLUDES TDEFAULTS DEBUGDIR
   } | write_mk gnu/lib/libbfd/arch/$MACHINE_ARCH/defs.mk

   write_c gnu/lib/libbfd/arch/$MACHINE_ARCH/bfd.h \
      <$_TMPDIR/bfd/bfd.h

   write_c gnu/lib/libbfd/arch/$MACHINE_ARCH/bfdver.h \
      <$_TMPDIR/bfd/bfdver.h

   {
      cat $_TMPDIR/bfd/config.h
   } | write_c gnu/lib/libbfd/arch/$MACHINE_ARCH/config.h
}
```

gnu/usr.bin/binutils/ld/Makefile

```
.include <bsd.own.mk>

TOP=            ${NETBSDSRCDIR}/gnu

.include "${.CURDIR}/../common/Makefile.inc"
.include "${.CURDIR}/arch/${MACHINE_ARCH}/defs.mk"

PROG=           ld
SRCS=           ${G_OFILES:.o=.c}
CPPFLAGS+=      -I${.CURDIR}/arch/${MACHINE_ARCH} \
                -I${DIST}/ld \
                -DDEFAULT_EMULATION=\"${G_EMUL}\" \
                -DSCRIPTDIR=\"/usr/share\" \
                -DTARGET=\"${G_target_alias}\" \
                -DBINDIR=\"/usr/bin\" \
                -DTOOLBINDIR=\"/usr/bin\"

LDADD=          -lintl
DPADD=          ${LIBINTL}

(snip)
```

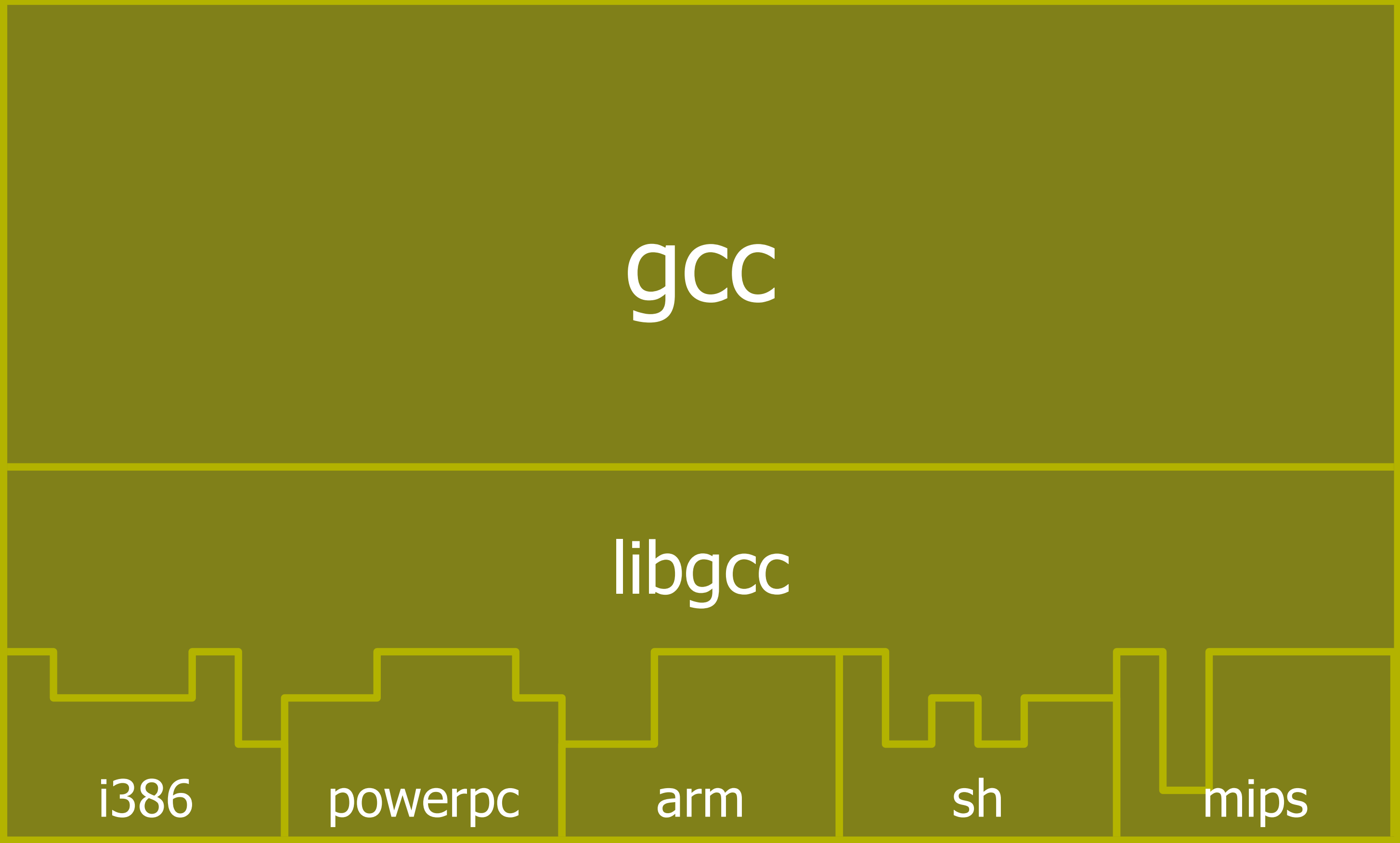
GCC

- gnu/dist/gcc4/gcc/config/*/*
- Build procedure
  - configure
  - bootstrap libgcc
  - xgcc
  - libgcc
- ${CC} <- configure && gmake
- /usr/bin/gcc <- nbmake-${XXX}
- Crosstools

libgcc

- The GCC low-level runtime library
- http://gcc.gnu.org/onlinedocs/gccint/Libgcc.html
- To perform some operation that is too complicated to emit inline code for

libgcc



libgcc – Installed Files

- /usr/lib/libgcc{,_p,_pic}.a
  - Static library
- /usr/lib/libgcc_eh{,_p,_pic}.a
  - Exception handling routines
- /usr/lib/libgcc_s.so{,.1,.1.0}
  - For C++ programs (?)
  - When -shared-libgcc is passed
  - :
  - ???

libgcc - Functions

- Integer library routines
- Soft float library routines
- Decimal float library routines
- Fixed-point fractional library routines
- Exception handling routines
- Miscellaneous routines

libgcc - Integer Library Routines

- Arithmetic functions
  (arithmetic shift, logical shift, quotient, reminder, ...)
- Comparison functions
  (signed comparison, unsigned comparison, ...)
- Trapping arithmetic functions
  (absolute value, sum, product, ...)
- Bit operations
  (leading 0-bits, trailing 0-bits, least significant 1-bit, ...)

libgcc - Soft Float Library Routines

- Arithmetic functions
  (sum, difference, product, ...)
- Conversion functions
  (extend, truncate, convert to signed int, ...)
- Comparison functions
  (a <=> b, is NaN, greater than, ...)
- Other floating-point functions
  (raise a to power b, product of (a+ib) and (c+id), ...)

libgcc - Decimal Float Library Routines

- Arithmetic functions
- Conversion functions
- Comparison functions

libgcc - Fixed-point Fractional Library Routines

- Arithmetic functions
- Conversion functions
- Comparison functions

libgcc - Exception Handling Routines

- Stack unwinding (_Unwind_*, _register_frame_*, _deregister_frame_*, ...)

libgcc - Miscellaneous Routines

- Cache control functions

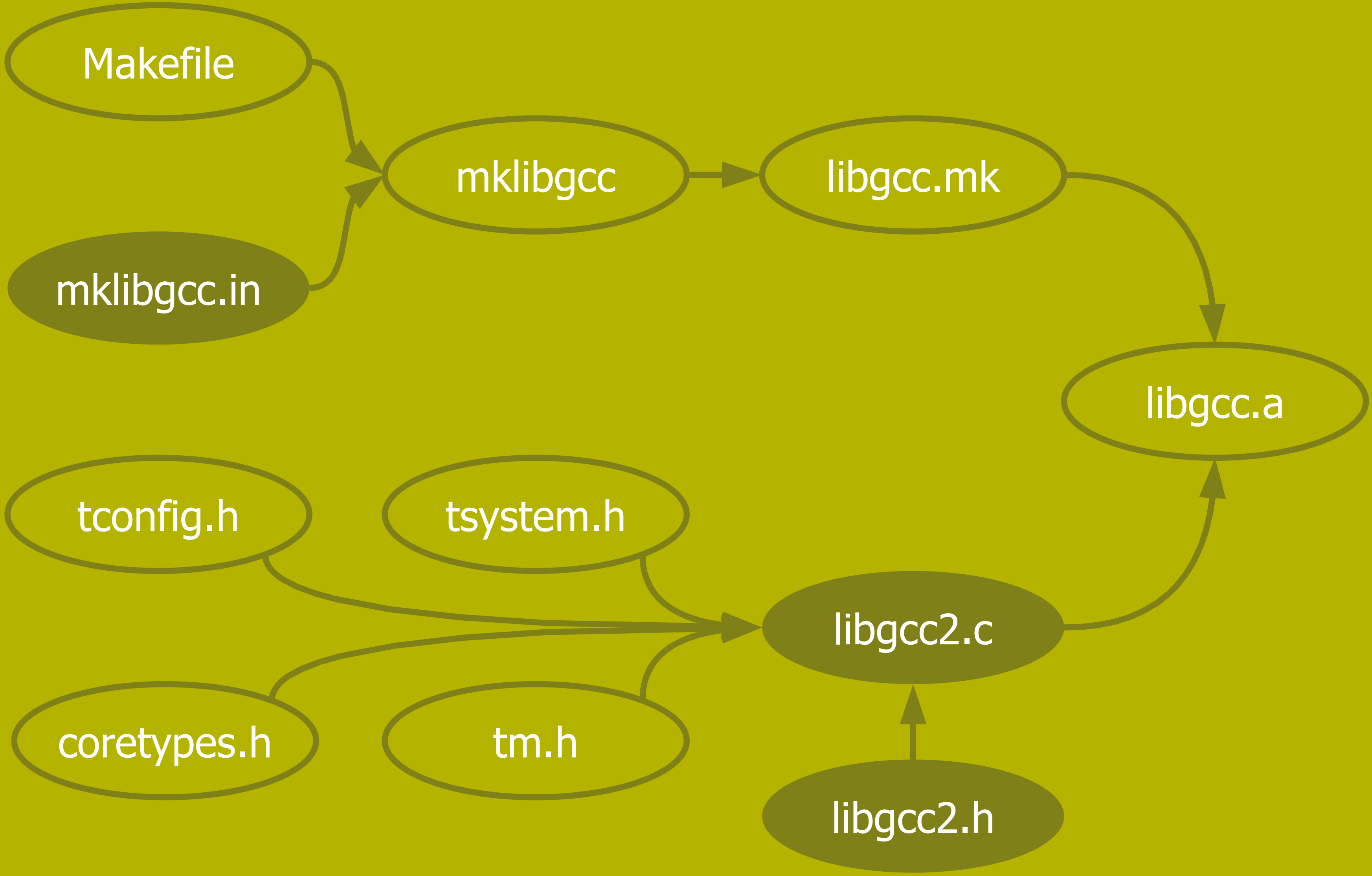
libgcc - Configuration

- Files
  - GNU Autoconf arguments
  - `gnu/dist/gcc4/gcc/config/*/*`
- Which functions (subset)
- Parameters
  - XXX

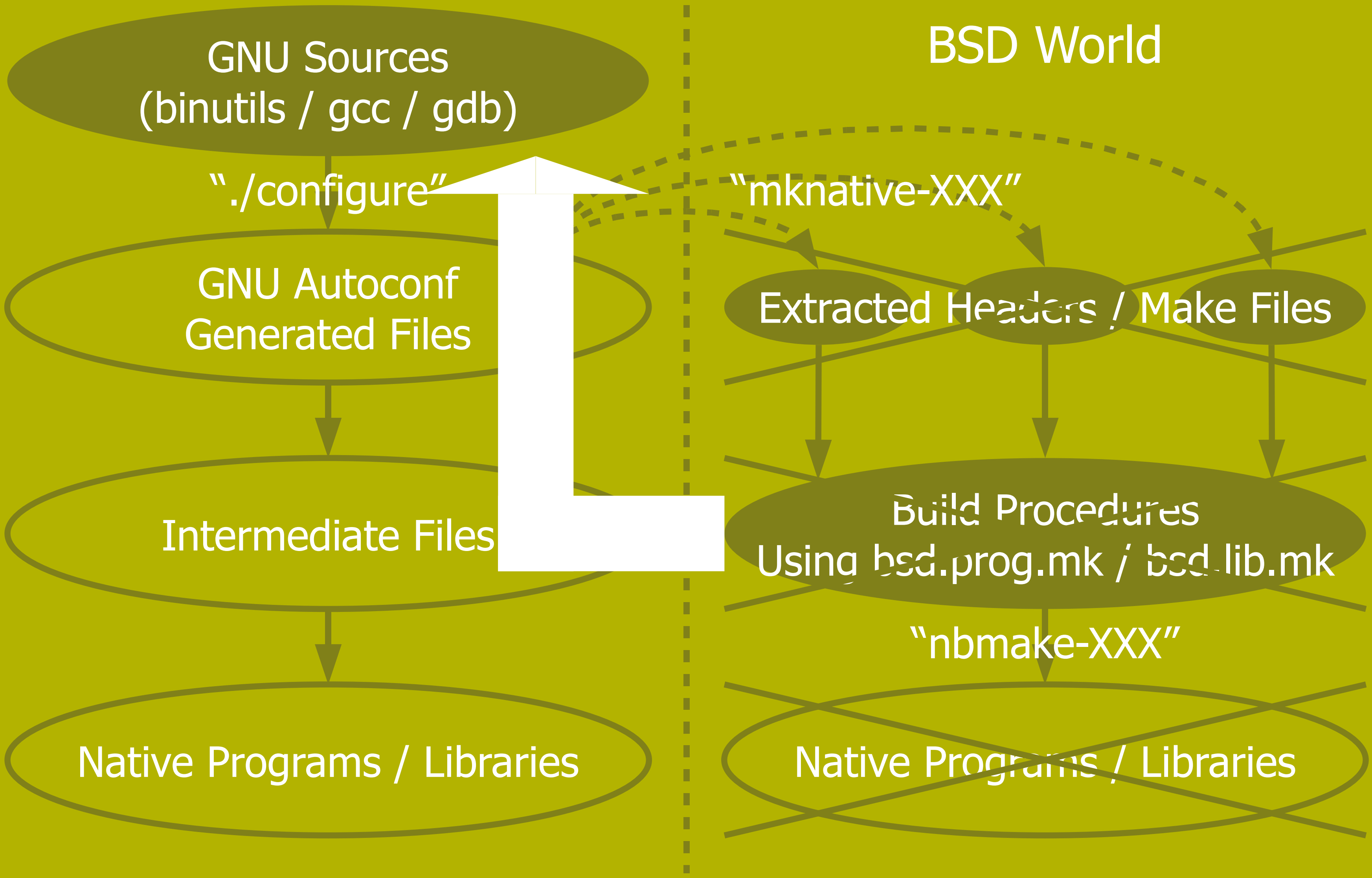
libgcc - Source Files

- gnu/dist/gcc4/gcc/Makefile.in
- gnu/dist/gcc4/gcc/libgcc-std.ver
- gnu/dist/gcc4/gcc/libgcc2.c
- gnu/dist/gcc4/gcc/libgcc2.h
- gnu/dist/gcc4/gcc/mklibgcc.in

libgcc - Build Dependency



Move Back to GNU Build Framework?



Conclusion

- mknative is great
  - ... while it works
- Problems
  - Hard to maintain (esp. when updating versions)
  - ... so, "don't"
- **"There is no royal road to mknative"**
- See libgcc symbol errors?
  - Doubt mknative!
- **Everyone should use "HOST_CFLAGS=-g"!!!!!1!**

Future

- Why do we use <bsd.{prog,lib}.mk>?
- (Internal) interface of /usr/share/mk/*
- Clarify GCC build procedure
  - Crosstools
- pkgsrc/cross?