Rethinking Basic Primitives
For Store Based Systems

Farid Zakaria¹, Tom Scogland² & Carlos Maltzahn¹

¹ - University of California Santa Cruz
² - Lawrence Livermore National Laboratory
“Eelco opening up pandora's box and releasing NixOS” - https://labs.openai.com/s/Co1EBnD7bfriJItGw2rQP6YM
Shared libraries

The cause of, and solution to, all of life’s (security, maintenance and packaging) problems.
nix-store --query --include-outputs --graph $(nix-store --query --deriver $(which ruby)) > /tmp/ruby_full_closure.dot
sfdp -x -Goverlap=prism -Tpng /tmp/ruby_full_closure.dot > output.png
The Rise of Convention

/lib /etc /bin

Filesystem Hierarchy Standard
Store Model - An alternative to the FHS model

```bash
$ patchelf --print-rpath ruby | tr ':\n' '
' /nix/store/yyhykbfjxj6jfc62x4f9a2awfbch605fd-ruby-2.6.6/lib
/nix/store/6hhqb94ilrdmh0cx5bhdqb92m6bqkvdj-zlib-1.2.11/lib
/nix/store/90illc73xfsh33d06daq6d41njs8yh66-glbc-2.32-37/
```
Problem Solved?
$ libtree /usr/bin/dbwrap_tool

libpopt-samba3-samba4.so [runpath]
  |  libcli-smb-common-samba4.so [runpath]
  |    |  libiov-buf-samba4.so [runpath]
  |    |  libsmb-transport-samba4.so [runpath]
  |    |  liblibsamba-sockets-samba4.so [runpath]
  |    |  libgensec-samba4.so [runpath]
  |    |  libsamba-modules-samba4.so [runpath]
  |    |    |  libsamba-util.so.0 [default path]
  |    |    |  libtalloc.so.2 [default path]
  |    |    |  libsamba-errors.so.1 [default path]
  |    |  libsamba-debug-samba4.so not found
  libdbwrap-samba4.so [runpath]
  libutil-tdb-samba4.so [runpath]
  |  libsamba-debug-samba4.so [runpath]
  |    |  libpopt.so.0 [default path]
  |    |  libtalloc.so.2 [default path]
  |    |  libsamba-errors.so.1 [default path]
  |    |  libsmsmbconf.so.0 [default path]
  |    |  libsamba-util.so.0 [default path]
Issue #2

Two options:
- dirA:dirB
- dirB:dirA

No way to get desired libraries using the existing coarse search controls (RUNPATH or LD_LIBRARY_PATH)
Potentially up to 3600 unnecessary filesystem calls

*In practice ~1800 filesystem calls.
Idea #1

Shrinkwrap: Flatten the graph and make absolute

https://github.com/fzakaria/shrinkwrap
Idea #2

Nix-Harden-Needed: Bubble up the world explicitly

https://github.com/fzakaria/nix-harden-needed

- Set the SONAME to the explicit /nix/store path
- This will get propagated automatically
- No need for RUNPATH anymore!

https://stoppels.ch/2022/08/04/stop-searching-for-shared-libraries.html
Idea #3

Do we still need to use ELF or it's fundamental concepts?

https://github.com/fzakaria/<TBD>

/nix/store/
  └── 9lz9yc6zgmc0vlqm2ipcpkjlmbi51vv-hello-1.9/
      └── bin/
          └── hello.rodata
          └── hello
          └── hello.strings
          └── hello.needed
          └── hello.dynsym
Idea #4

Nix needs its own dynamic linker

https://github.com/fzakaria/<TBD>

“An oil painting of a dynamic linker for Nix” - https://labs.openai.com/s/y5ZGjSuWBX3S5e7zZgfcQoh6
Thank you

Questions?