Working around POSIX's faults

Improving the reliability of Linux named services (NSS) for large institutions

Jamie Wilkinson <jaq@google.com>
V Hoffman <vasilios@google.com>

POSIX 1003.1-2004

The API

```
get*nam()

get*id()

get*ent()
```

API gets called all the time!

```
login: jaq
Password: ****
% ls -l
total 1
drwx------  2 jaq users  4096 Jan  8 10:20
Desktop/
% host linux.conf.au
linux.conf.au has address 221.133.213.165
% sudo -i
Password: ****
% cd ~<TAB>
```

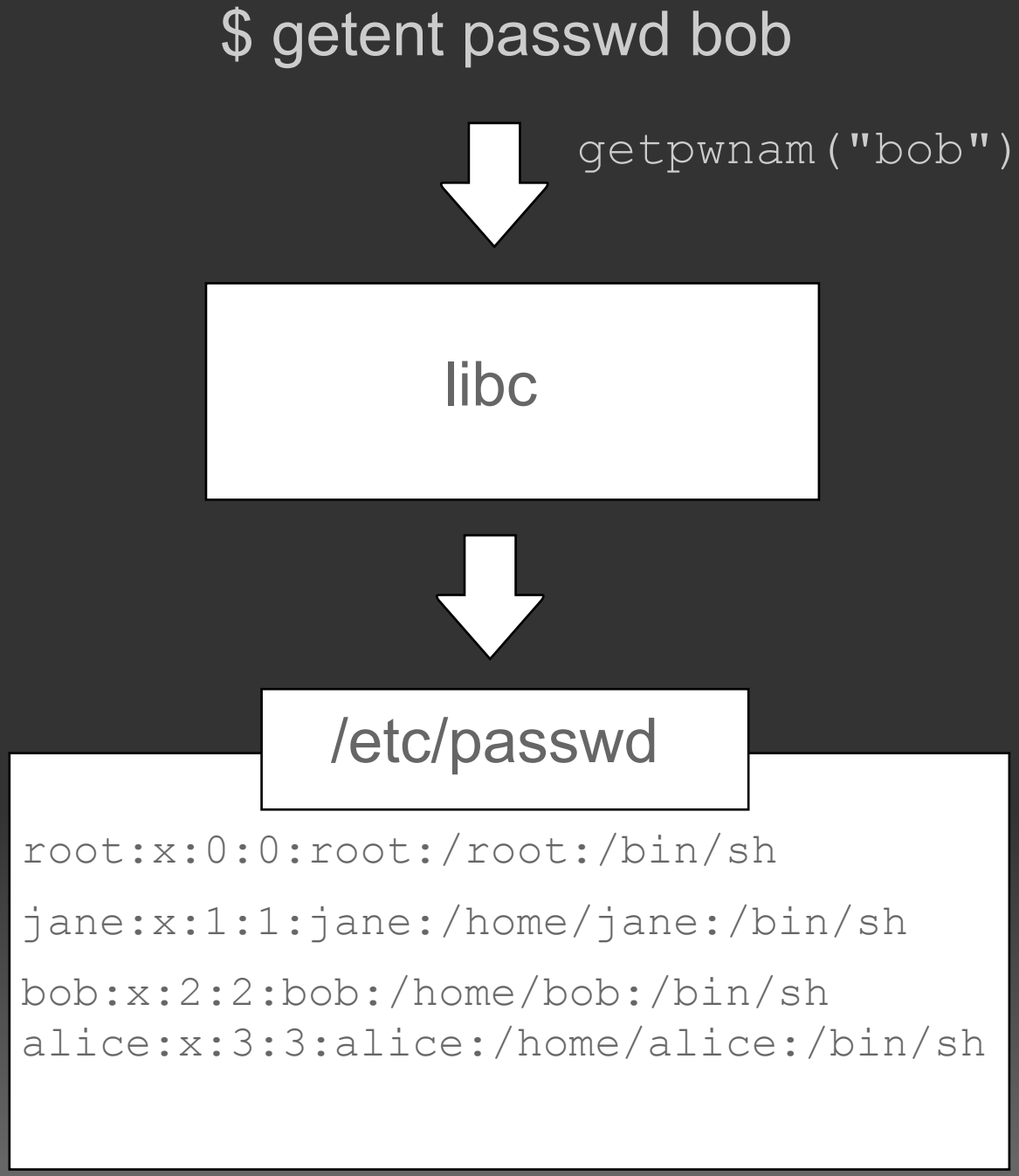
...where does the data come from?

Databases were plain text files

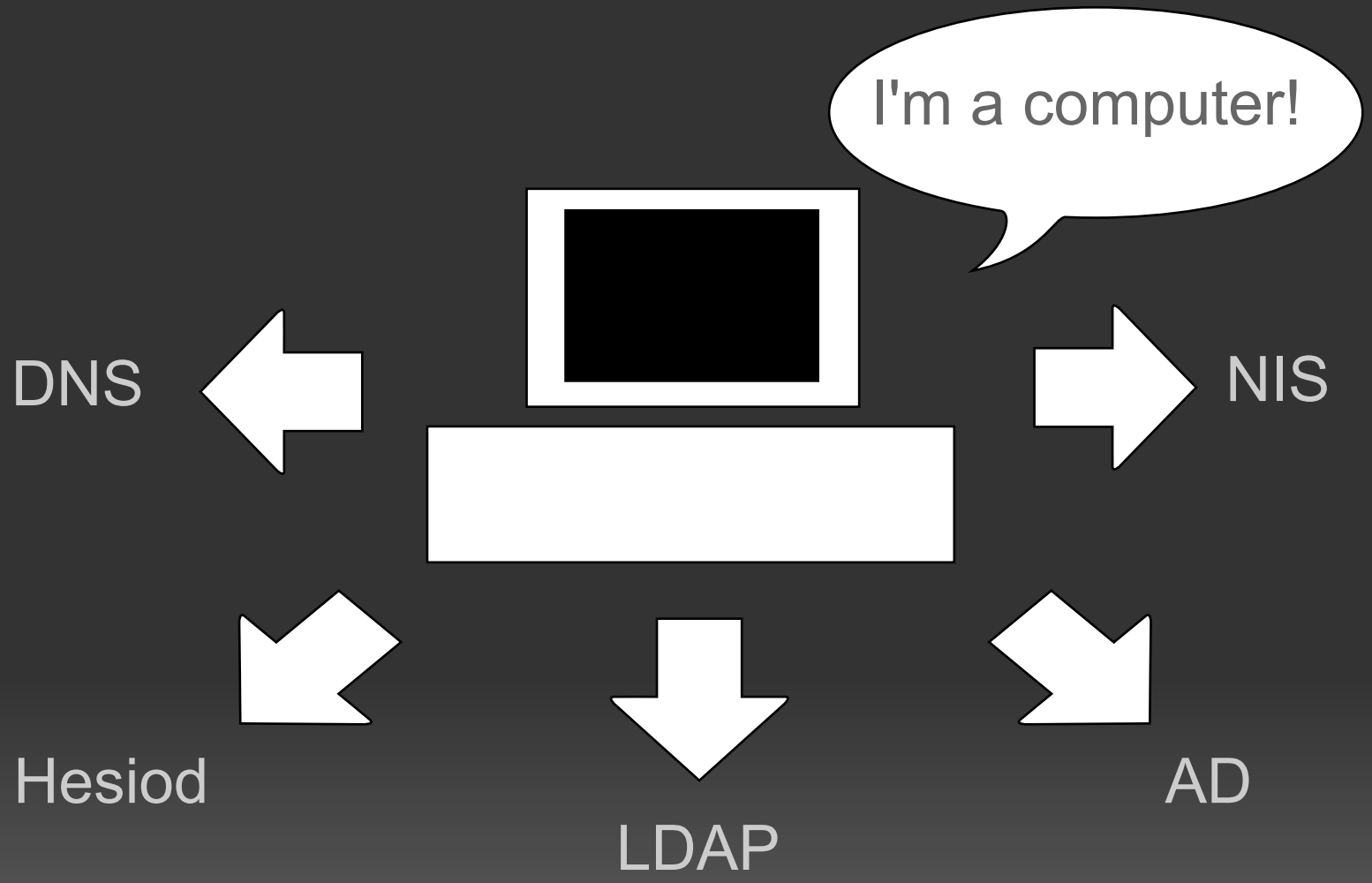
```
root:x:0:0:root:/root:/bin/bash
alice:x:101:100:alice:/home/alice:/usr/bin/vi
bob:x:102:100:bob:/home/bob:/usr/bin/emacs
ed:x:103:100:ed:/home/ed:/bin/ed
leet:x:103:100:leet:/home/leet:/dev/kmem
```

... then resources started to centralise!

A Lookup


$ getent passwd bob
getpwnam("bob")
libc
/etc/passwd
root:x:0:0:root:/root:/bin/sh
jane:x:1:1:jane:/home/jane:/bin/sh
bob:x:2:2:bob:/home/bob:/bin/sh
alice:x:3:3:alice:/home/alice:/bin/sh

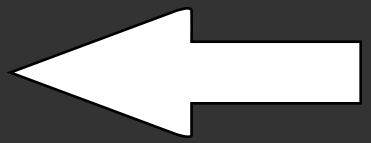
Want data from other sources



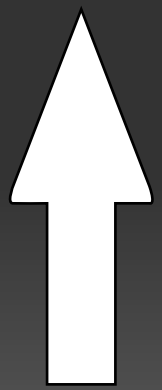
The solution: Name Service Switch

```
# /etc/nsswitch.conf

passwd: compat files
groups: compat files
shadow: compat files
hosts:  files dns
```

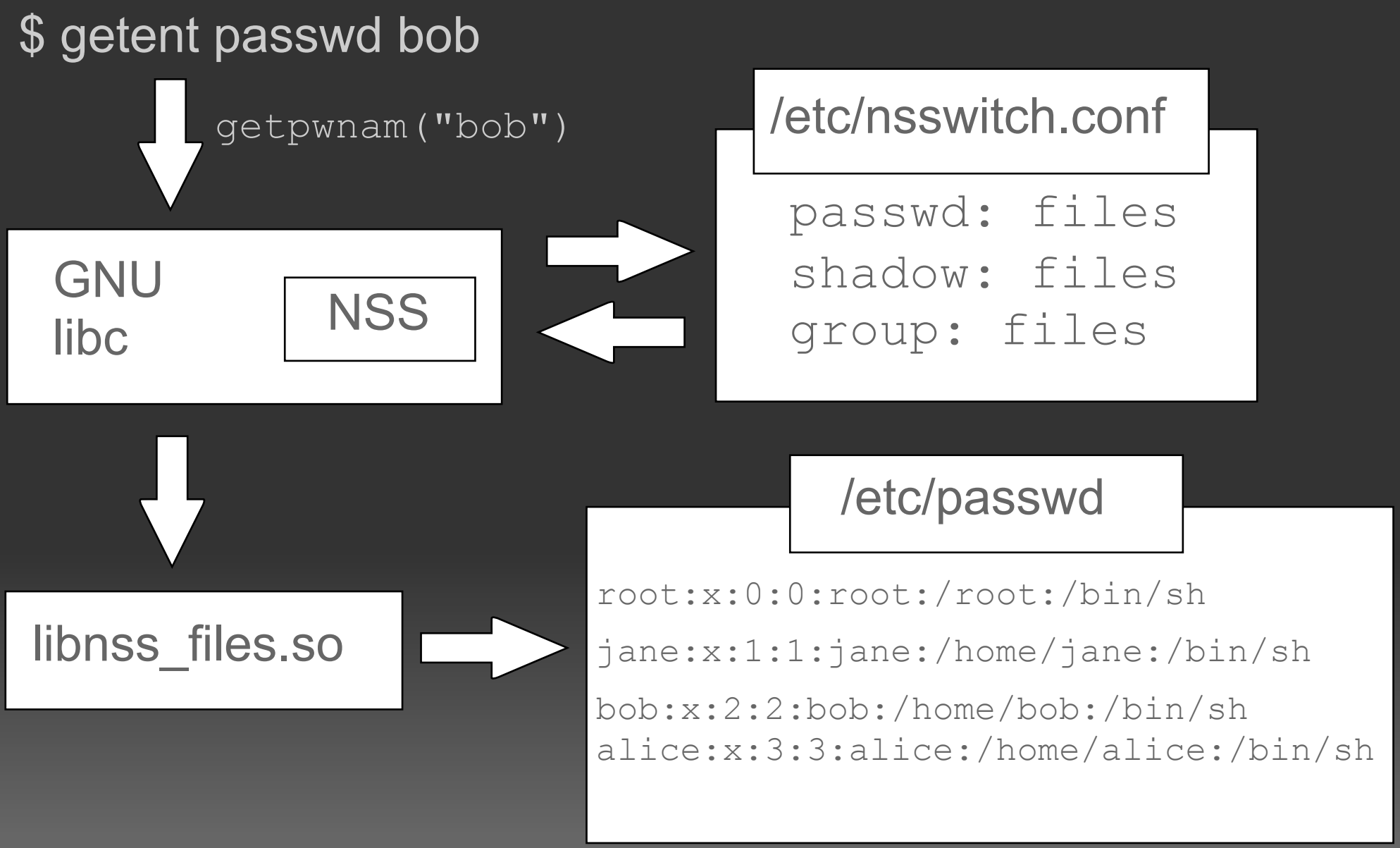


location of data

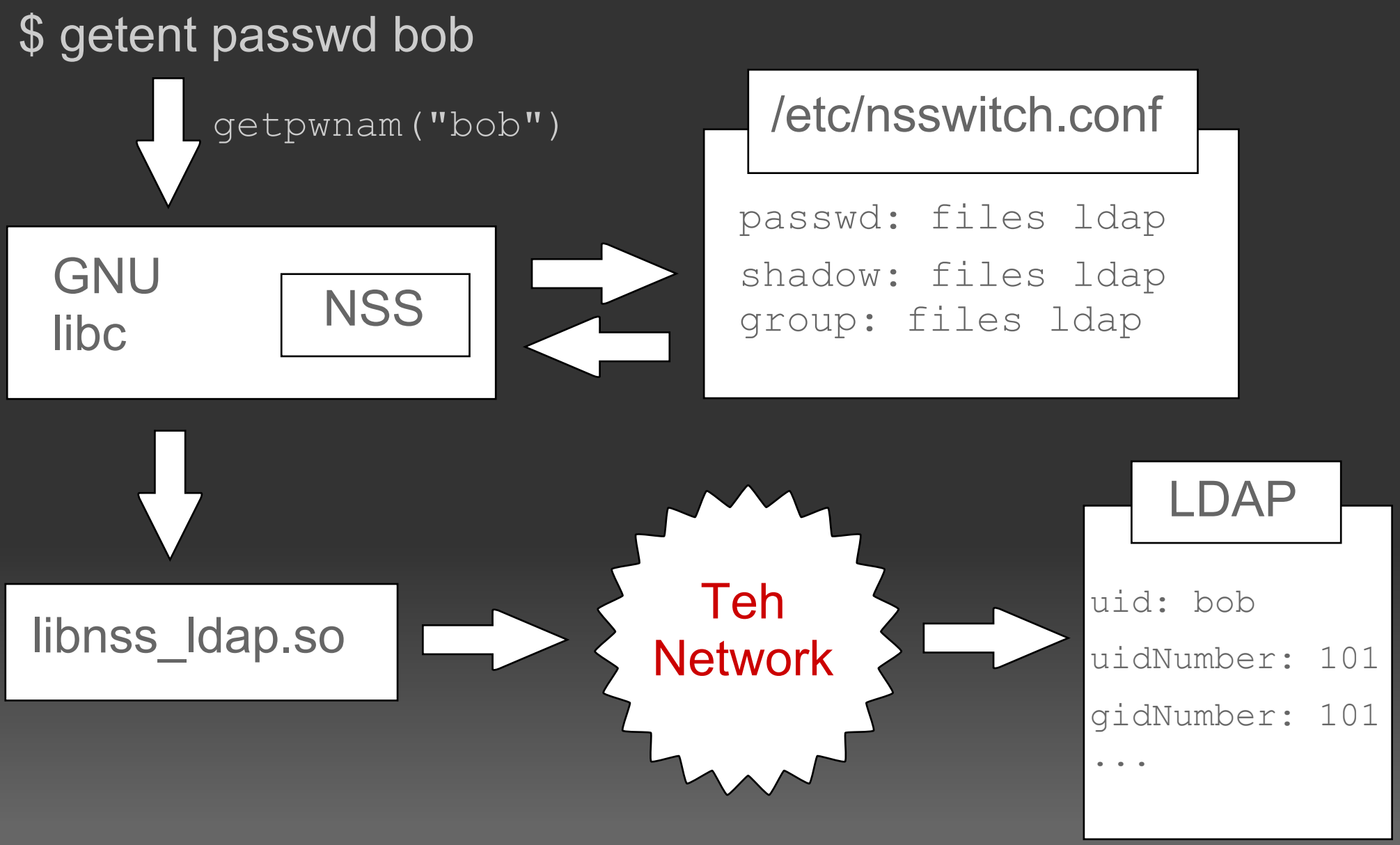


type of data

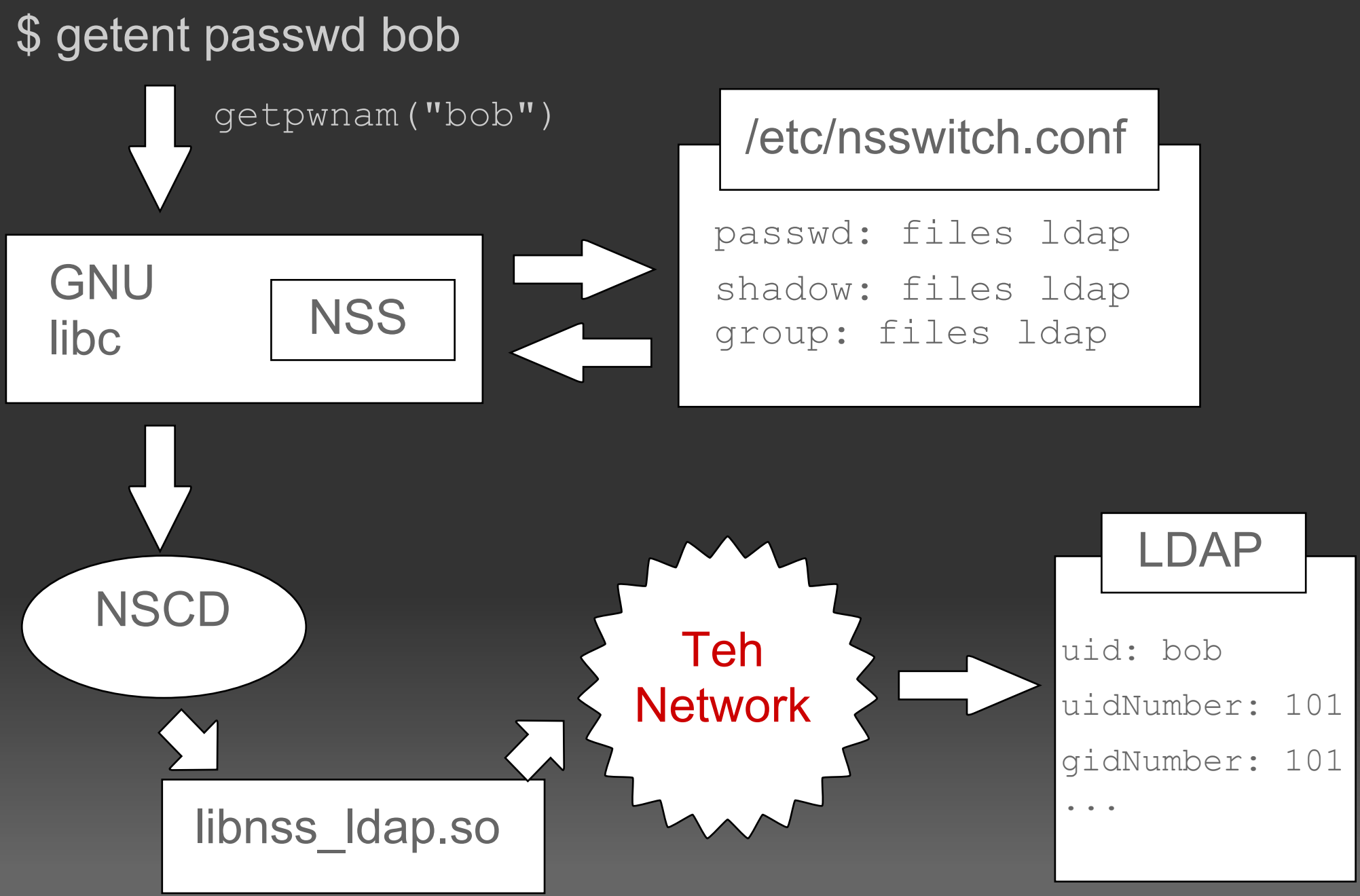
NSS


$ getent passwd bob
getpwnam("bob")
GNU
libc
NSS
/etc/nsswitch.conf
passwd: files
shadow: files
group: files
libnss_files.so
/etc/passwd
root:x:0:0:root:/root:/bin/sh
jane:x:1:1:jane:/home/jane:/bin/sh
bob:x:2:2:bob:/home/bob:/bin/sh
alice:x:3:3:alice:/home/alice:/bin/sh

NSS + LDAP


$ getent passwd bob
getpwnam("bob")
GNU
libc
NSS
/etc/nsswitch.conf
passwd: files ldap
shadow: files ldap
group: files ldap
libnss_ldap.so
Teh
Network
LDAP
uid: bob
uidNumber: 101
gidNumber: 101
...

NSS + LDAP + NSCD



NSS is fast and never fails

... if only we had `EAGAIN`

Effects of failure on NSS

Access

Behaviour

Speed

... worse, it's often transient!

General causes of failure

Networks, services are unreliable

Reliability is expensive

... at the end of the day, NSS still expects 100% reliability

Performance impact on the user

0.1 second : instantly responsive

1 second : thought interrupted

... how do you increase the speed of light?

Miller, R. B. (1968). Response time in man-computer conversational transactions. *Proc. AFIPS Fall Joint Computer Conference* **Vol. 33**, 267-277.

299,792,458 m/s

Teh Network

Lots of network traffic

10,000 users
1,000 groups

= 6 MB for passwd database
e.g. ls -l /home, cd ~<TAB>

= 1 MB for 10k member group

... more than 0.1 seconds!

Volume of queries

~7000 LDAP queries/day per host

Uneven Traffic

Peak Traffic

... for a small controlled LAN you may not see this enough to care :-)

If I had a nickel for every packet

(A nickel is just under 6 australian cents.)

API inefficient

Uncacheable

TTL

Software is hard

...and dammit Jim, I'm a sysadmin, not a programmer!

Requirements for a solution

Goodbye Network

Reduce Complexity

Persistance

SLA

... but I'm just a lowly tape monkey!

That 70s Show™

```
root:x:0:0:root:/root:/bin/bash
alice:x:101:100:alice:/home/alice:/usr/bin/vi
bob:x:102:100:bob:/home/bob:/usr/bin/emacs
ed:x:103:100:ed:/home/ed:/bin/ed
leet:x:103:100:leet:/home/leet:/dev/kmem
```

... look familiar?

Cron and a Script

```
*/5 * * * * ldapsearch | awk > /etc/passwd
```

NSS Cache

```
# /etc/nsscache.conf

[DEFAULT]

# Default NSS data source module name
source = ldap

# Default NSS data cache module name
cache = nssdb

# NSS maps to be cached
maps = passwd, group, shadow
```

TODO

Automount Support

Performance

Local Rewrites

Pay attention to code.google.com

Ponies

Questions?

http://code.google.com/p/nsscache