Unifying interfaces

Kevin Lynagh
SpeakerConf Barcelona
May 2014

Pop Quiz

Pop Quiz

List all git tags

```
$ git tag
```

Pop Quiz

List all git remotes

```
$ git remote -v
```

Pop Quiz

List all git branches

```
$ git branch -a
```

Pop Quiz

List all git commit SHAs

```
$ git rev-list HEAD
```

Pop Quiz

List folder

```
$ ls
```

Pop Quiz

List current git root

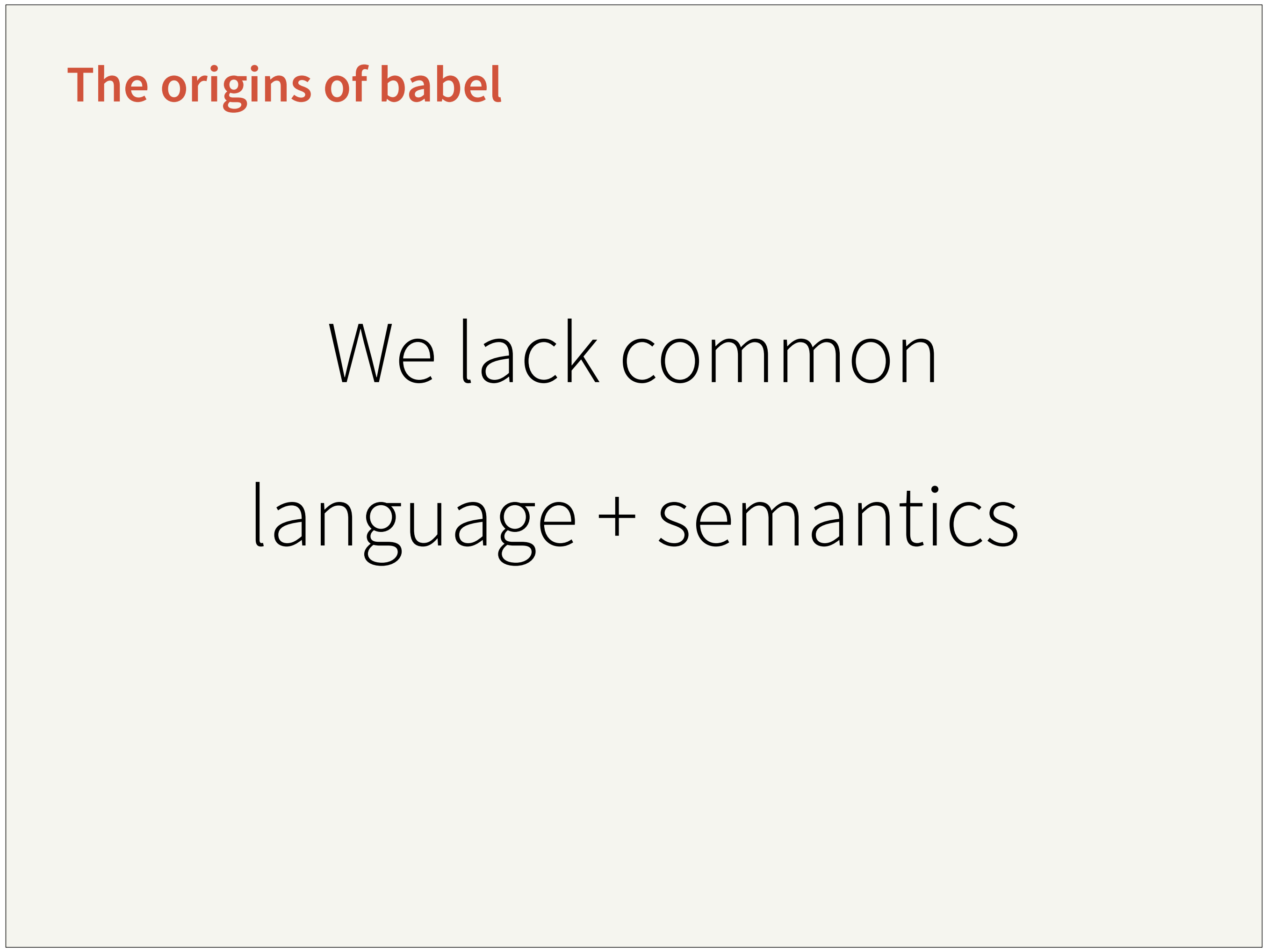
```
$ git ls-tree HEAD --name-only
```

This sucks…

- Poor habituation
- It’s trivia
- Similar semantics should have similar interfaces

Overview

1. Why are things this way?
2. How else could it be?



The origins of babel

We lack common

language + semantics

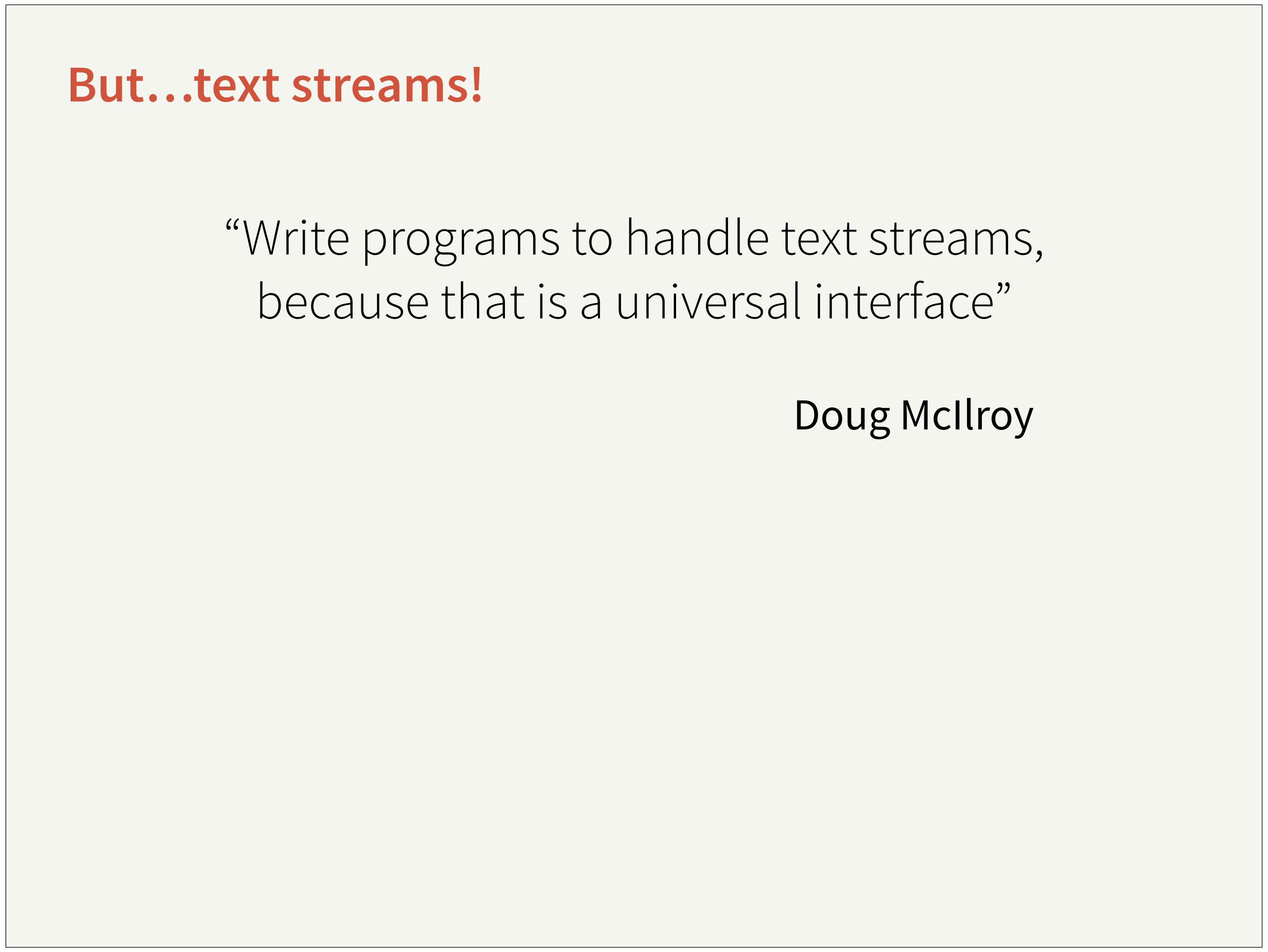
Babel results from lack of common language/semantics.

Without explicit protocols, identical semantics are fractured across dozens of incantations.

It's not easy to talk about protocols vs. implementations in *nix.

Without a name for the idea (e.g., Iterable), it's very easy to ignore the semantics and focus on just the implementation.

And if your only concern is the implementation of a single component, at no point are you concerned with the overall system cohesion.



But…text streams!

"Write programs to handle text streams, because that is a universal interface"

Doug McIlroy

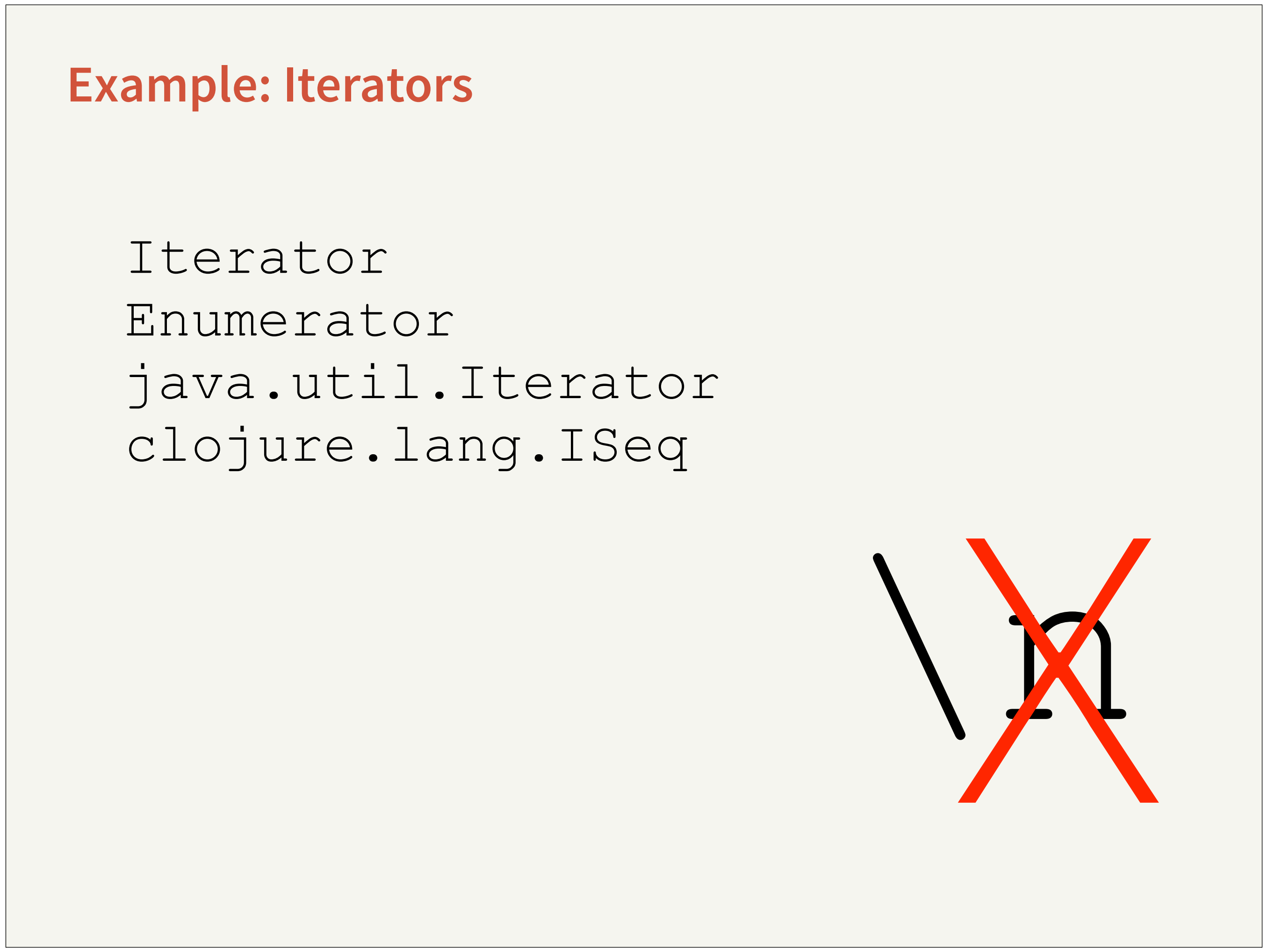
Humans have shared interface of sound waves: Just grunts.

There's no such thing as plain text: Everything has structure.
Keeping that structure implicit helps no one (we all just internalize inconsistent conventions and write shitty parsers).

Guess: most streams are more an artifact of implementation (e.g., tapes) rather than ideal semantics

Explicit protocols enable rich thought

- Consistent, usable interface
- Reuse + composition
- Smarter tooling



Example: Iterators

```
Iterator
Enumerator
java.util.Iterator
clojure.lang.ISeq
```

$\cancel{\backslash n}$

These iterate over the semantically-meaningful elements, not the implementation details of bytes

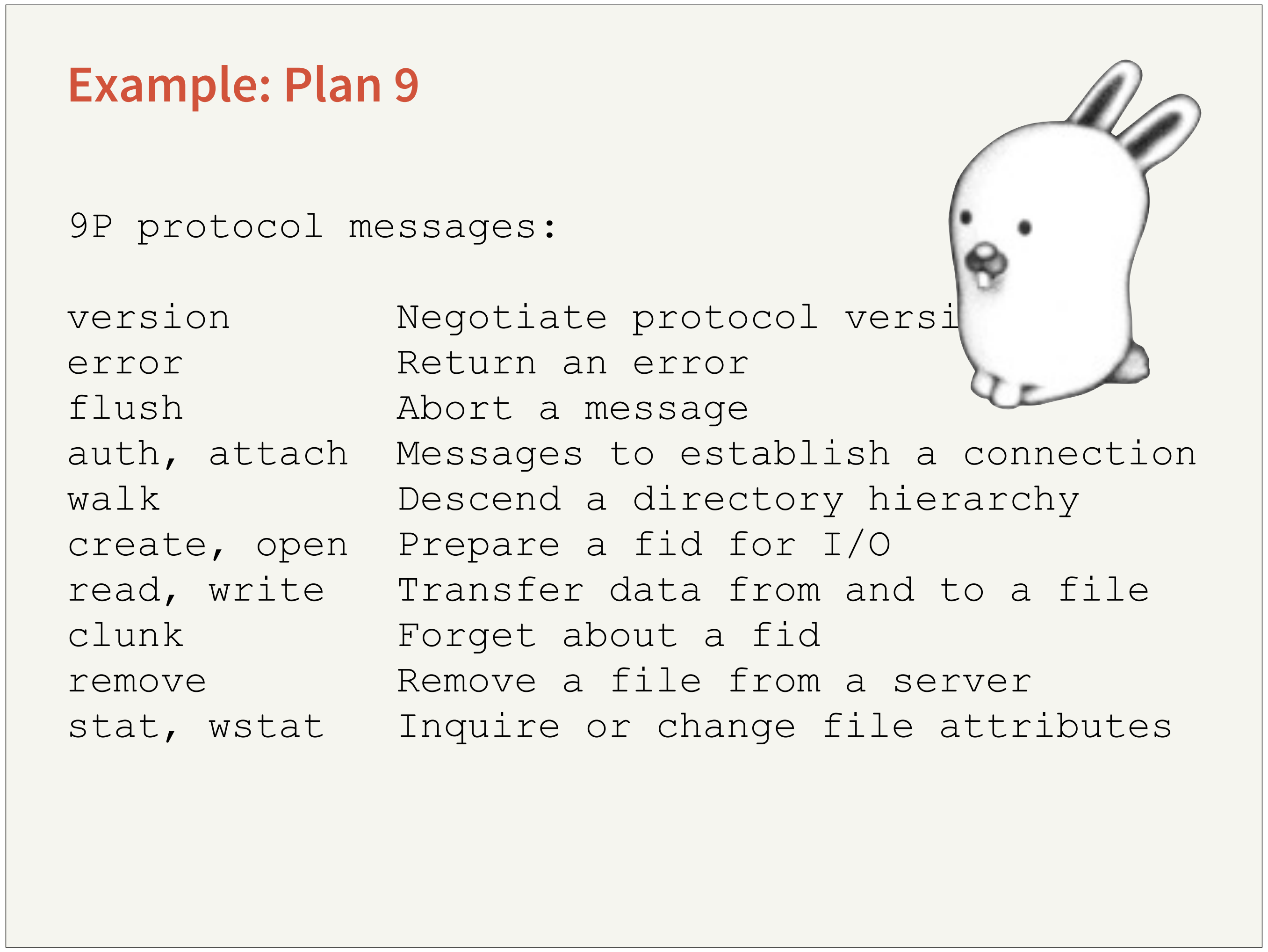
Example: Filesystems

File

Folder

+ files: which are just byte streams associated with fixed metadata (name, size, permissions)

+ directories: which are just collections of files with fixed metadata (name, permissions)



Example: Plan 9

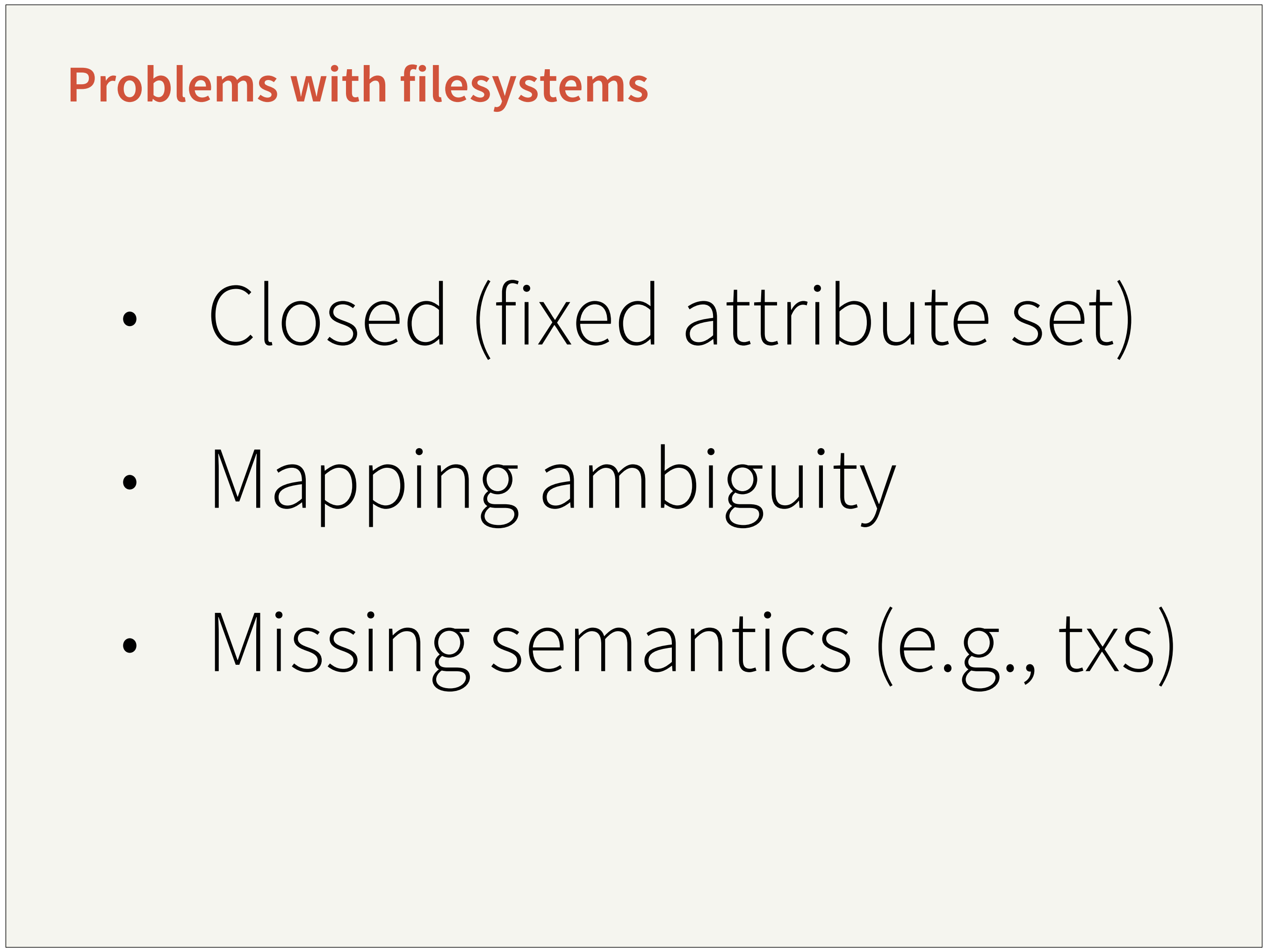
9P protocol messages:

version	Negotiate protocol versi
error	Return an error
flush	Abort a message
auth, attach	Messages to establish a connection
walk	Descend a directory hierarchy
create, open	Prepare a fid for I/O
read, write	Transfer data from and to a file
clunk	Forget about a fid
remove	Remove a file from a server
stat, wstat	Inquire or change file attributes

Plan 9 famously ran with the idea of having everything meet at the filesystem (actually, 9P protocol), with fruitful results.

Email clients would expose your inbox as a directory---you could move messages around with `mv` and delete them with `rm`.

If you wanted data on another computer you just mounted it---no need to fuck around with `scp`.



Problems with filesystems

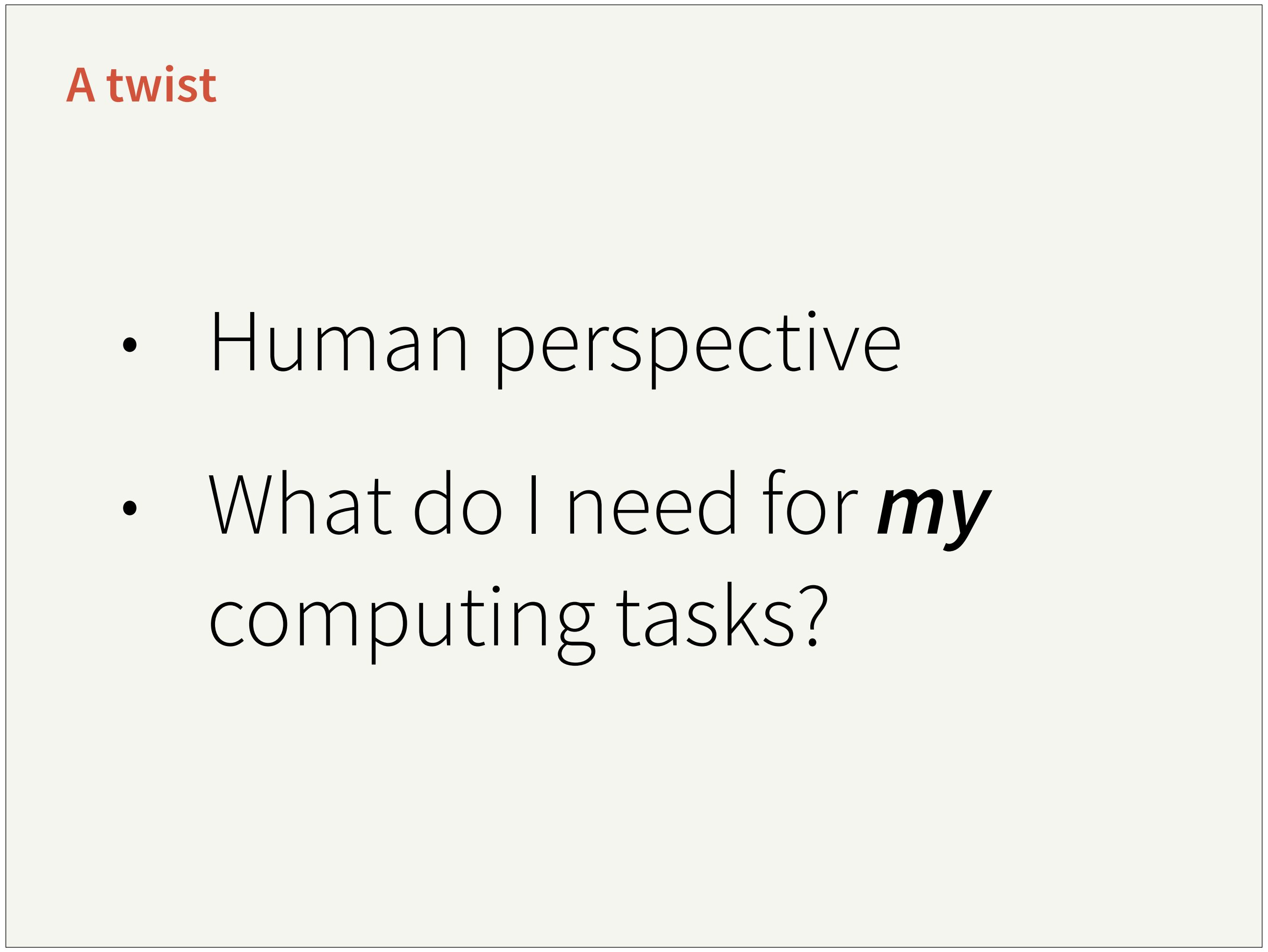
- Closed (fixed attribute set)
- Mapping ambiguity
- Missing semantics (e.g., txs)

E.g., how do you expose git tags?

+ a file with tag name whose contents is a 20-byte SHA?
  You can change the tag name with `mv` and the tag pointer with `cat new-sha > tagname`
  (Does the latter throw errors if the SHA does not resolve to a repo commit?)
+ a directory with tag name whose contents is the same contents as the associated commit?
  How do you change the tag or create new ones?

Crazy grand plan

1. uncover protocols
2. build a consistent editor/ environment (i.e., Emacs)



A twist

- Human perspective
- What do I need for ***my*** computing tasks?

Don’t care about machine efficiency or internal operations.

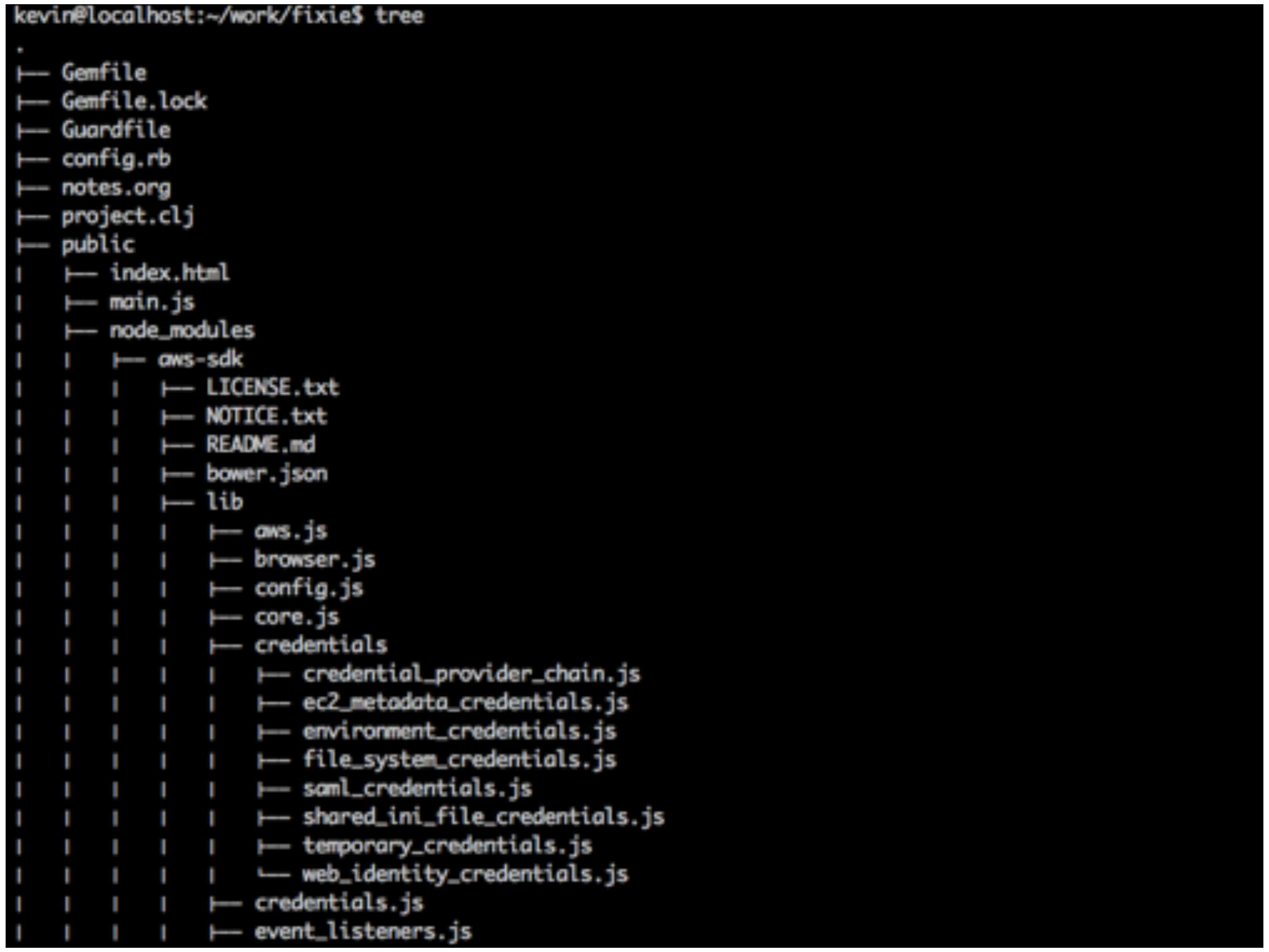
Care about habitability, learnability, and leverage.

How do I compute?

```
filesystem / Dropbox
terminal (execute processes)
email
calendar
web browsing
code editing
prose editing
```

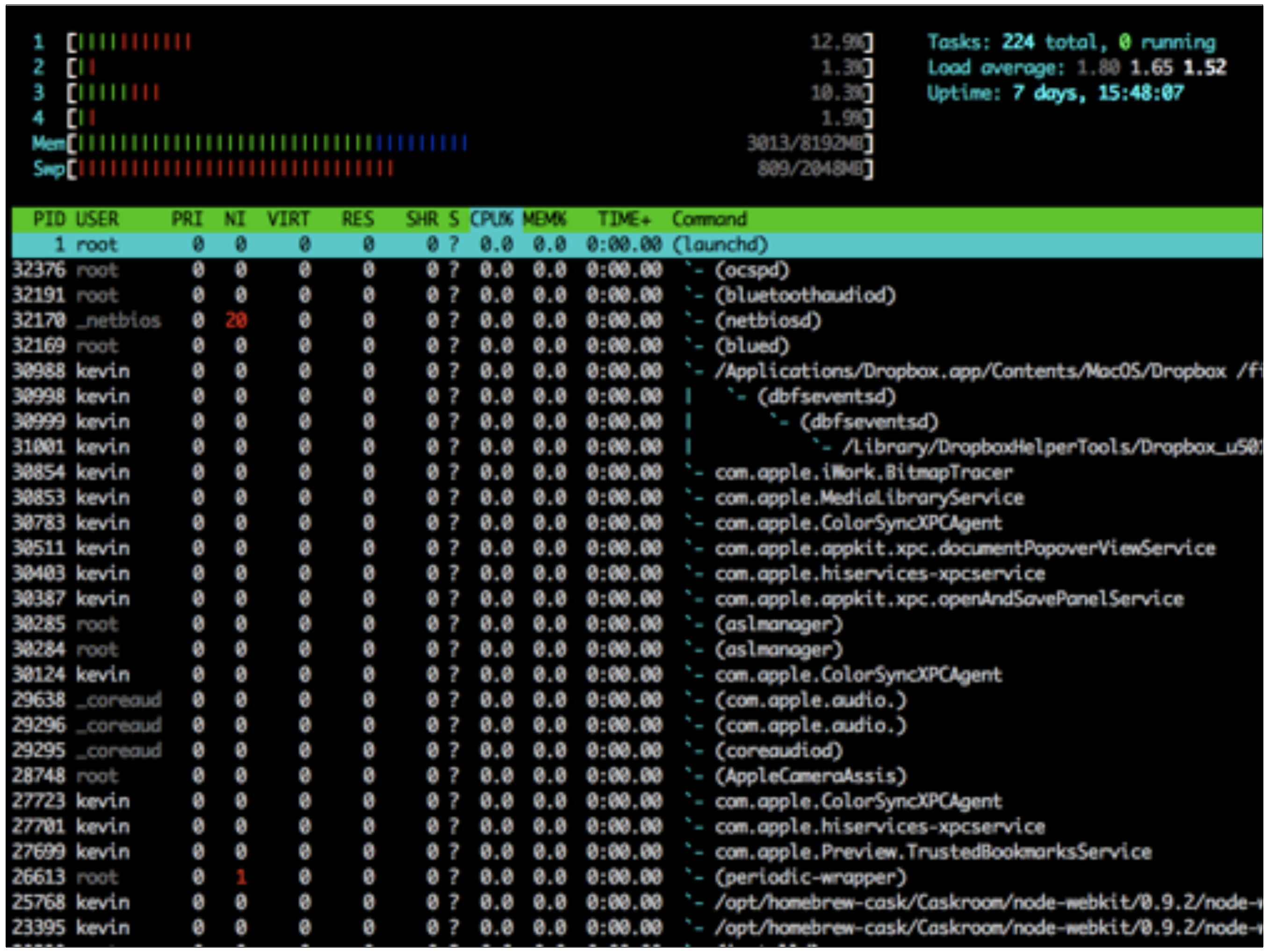
Similarities?

- Graphs of associative nodes



```
kevin@localhost:~/work/fixie$ tree
.
├── Gemfile
├── Gemfile.lock
├── Guardfile
├── config.rb
├── notes.org
├── project.clj
├── public
│   ├── index.html
│   ├── main.js
│   ├── node_modules
│   │   ├── aws-sdk
│   │   │   ├── LICENSE.txt
│   │   │   ├── NOTICE.txt
│   │   │   ├── README.md
│   │   │   ├── bower.json
│   │   │   ├── lib
│   │   │   │   ├── aws.js
│   │   │   │   ├── browser.js
│   │   │   │   ├── config.js
│   │   │   │   ├── core.js
│   │   │   │   ├── credentials
│   │   │   │   │   ├── credential_provider_chain.js
│   │   │   │   │   ├── ec2_metadata_credentials.js
│   │   │   │   │   ├── environment_credentials.js
│   │   │   │   │   ├── file_system_credentials.js
│   │   │   │   │   ├── saml_credentials.js
│   │   │   │   │   ├── shared_ini_file_credentials.js
│   │   │   │   │   ├── temporary_credentials.js
│   │   │   │   │   └── web_identity_credentials.js
│   │   │   │   ├── credentials.js
│   │   │   │   ├── event_listeners.js
```

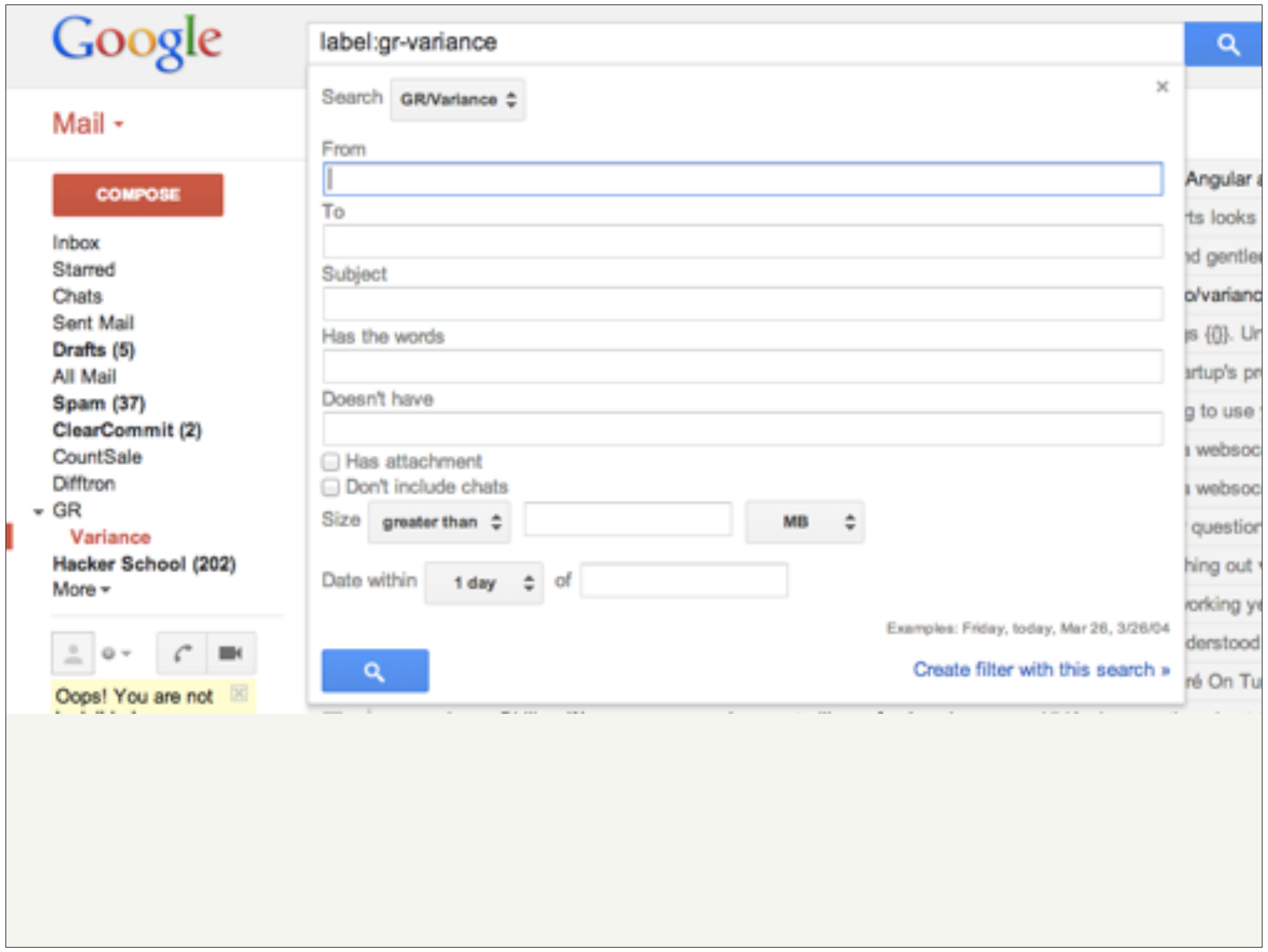
File/Dir: name, size, creation/modification times, tags, (edges to) children



```
  1  [||||||||||||                                     12.9%]   Tasks: 224 total, 0 running
  2  [||                                                1.3%]   Load average: 1.80 1.65 1.52
  3  [||||||||                                         10.3%]   Uptime: 7 days, 15:48:07
  4  [||                                                1.9%]
  Mem[||||||||||||||||||||||||||||||||||||||      3013/8192MB]
  Swp[|||||||||||||||||||||||||||||               809/2048MB]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
    1 root        0   0     0     0     0 ?  0.0  0.0  0:00.00 (launchd)
32376 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (ocspd)
32191 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (bluetoothaudiod)
32170 _netbios    0  20     0     0     0 ?  0.0  0.0  0:00.00  `- (netbiosd)
32169 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (blued)
30988 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- /Applications/Dropbox.app/Contents/MacOS/Dropbox /f
30998 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  |   `- (dbfseventsd)
30999 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  |       `- (dbfseventsd)
31001 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  |           `- /Library/DropboxHelperTools/Dropbox_u50
30854 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.iWork.BitmapTracer
30853 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.MediaLibraryService
30783 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.ColorSyncXPCAgent
30511 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.appkit.xpc.documentPopoverViewService
30403 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.hiservices-xpcservice
30387 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.appkit.xpc.openAndSavePanelService
30285 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (aslmanager)
30284 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (aslmanager)
30124 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.ColorSyncXPCAgent
29638 _coreaud    0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (com.apple.audio.)
29296 _coreaud    0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (com.apple.audio.)
29295 _coreaud    0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (coreaudiod)
28748 root        0   0     0     0     0 ?  0.0  0.0  0:00.00  `- (AppleCameraAssis)
27723 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.ColorSyncXPCAgent
27701 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.hiservices-xpcservice
27699 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- com.apple.Preview.TrustedBookmarksService
26613 root        0   1     0     0     0 ?  0.0  0.0  0:00.00  `- (periodic-wrapper)
25768 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- /opt/homebrew-cask/Caskroom/node-webkit/0.9.2/node-
23395 kevin       0   0     0     0     0 ?  0.0  0.0  0:00.00  `- /opt/homebrew-cask/Caskroom/node-webkit/0.9.2/node-
```

Processes: executable path, PID, args, (edge to) environment


label:gr-variance

Search: GR/Variance

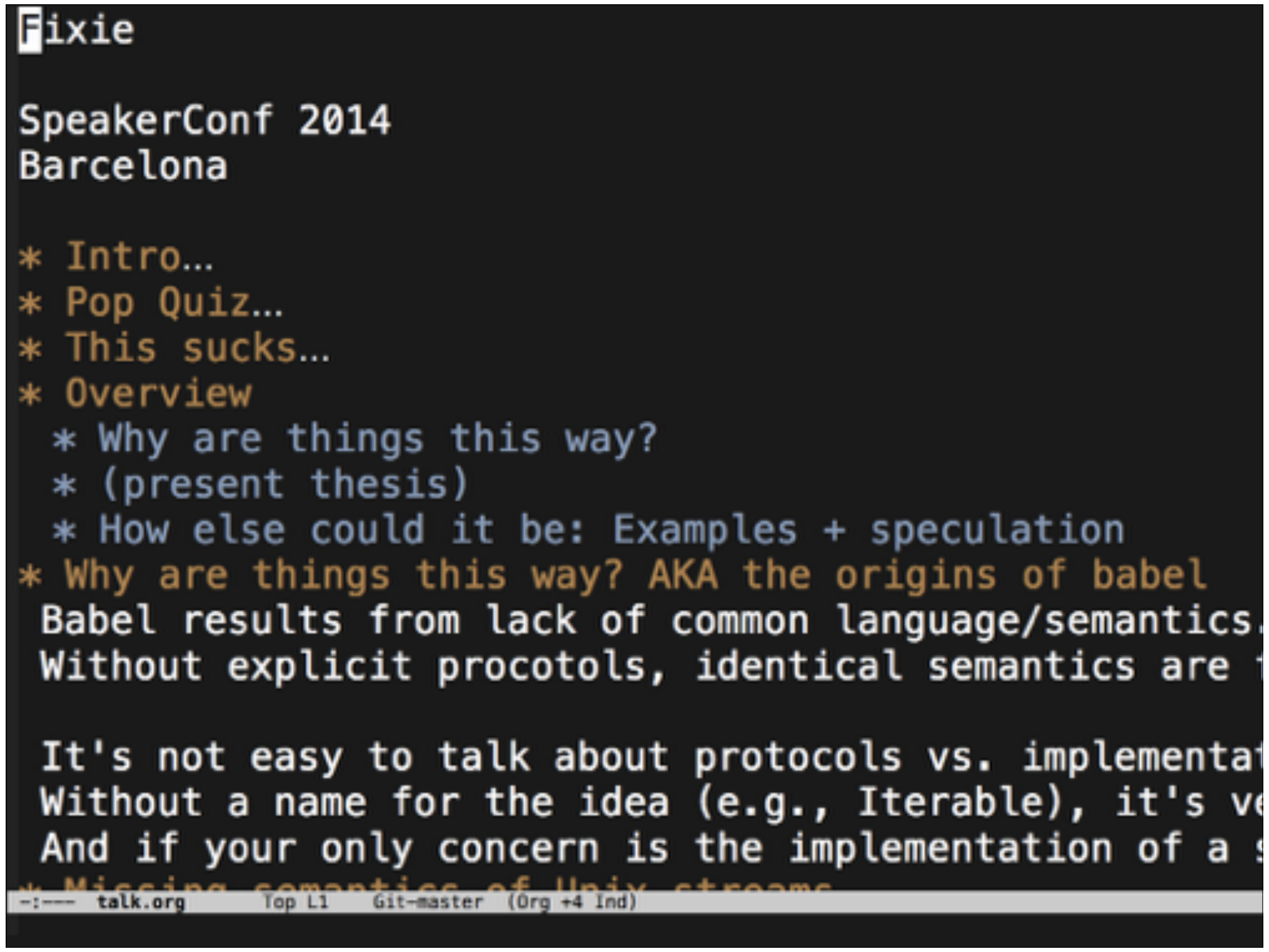
Size: greater than, MB

Date within: 1 day

Examples: Friday, today, Mar 26, 3/26/04

Oops! You are not

Email: to/cc/bcc, body, send date, (edges to) replies

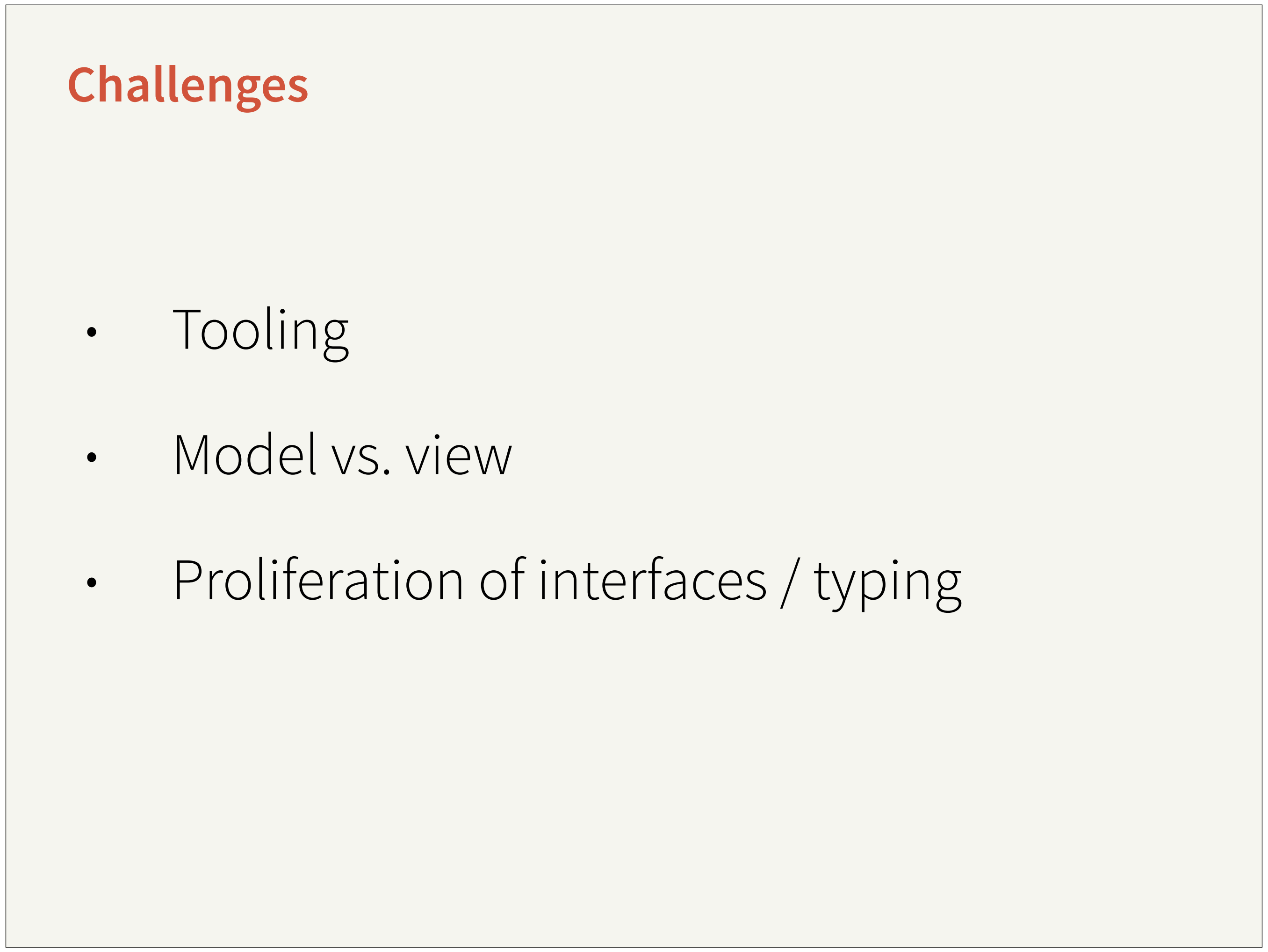


```
Fixie

SpeakerConf 2014
Barcelona

* Intro...
* Pop Quiz...
* This sucks...
* Overview
 * Why are things this way?
 * (present thesis)
 * How else could it be: Examples + speculation
* Why are things this way? AKA the origins of babel
 Babel results from lack of common language/semantics.
 Without explicit procotols, identical semantics are
 
 It's not easy to talk about protocols vs. implementat
 Without a name for the idea (e.g., Iterable), it's ve
 And if your only concern is the implementation of a
* Missing semantics of Unix streams
-:---  talk.org       Top L1     Git-master   (Org +4 Ind)
```

Sections, subsections, paragraphs, sentences



Challenges

- Tooling
- Model vs. view
- Proliferation of interfaces / typing

Tooling is not biggest challenge—the bar is very low already and shouldn’t be difficult to match.

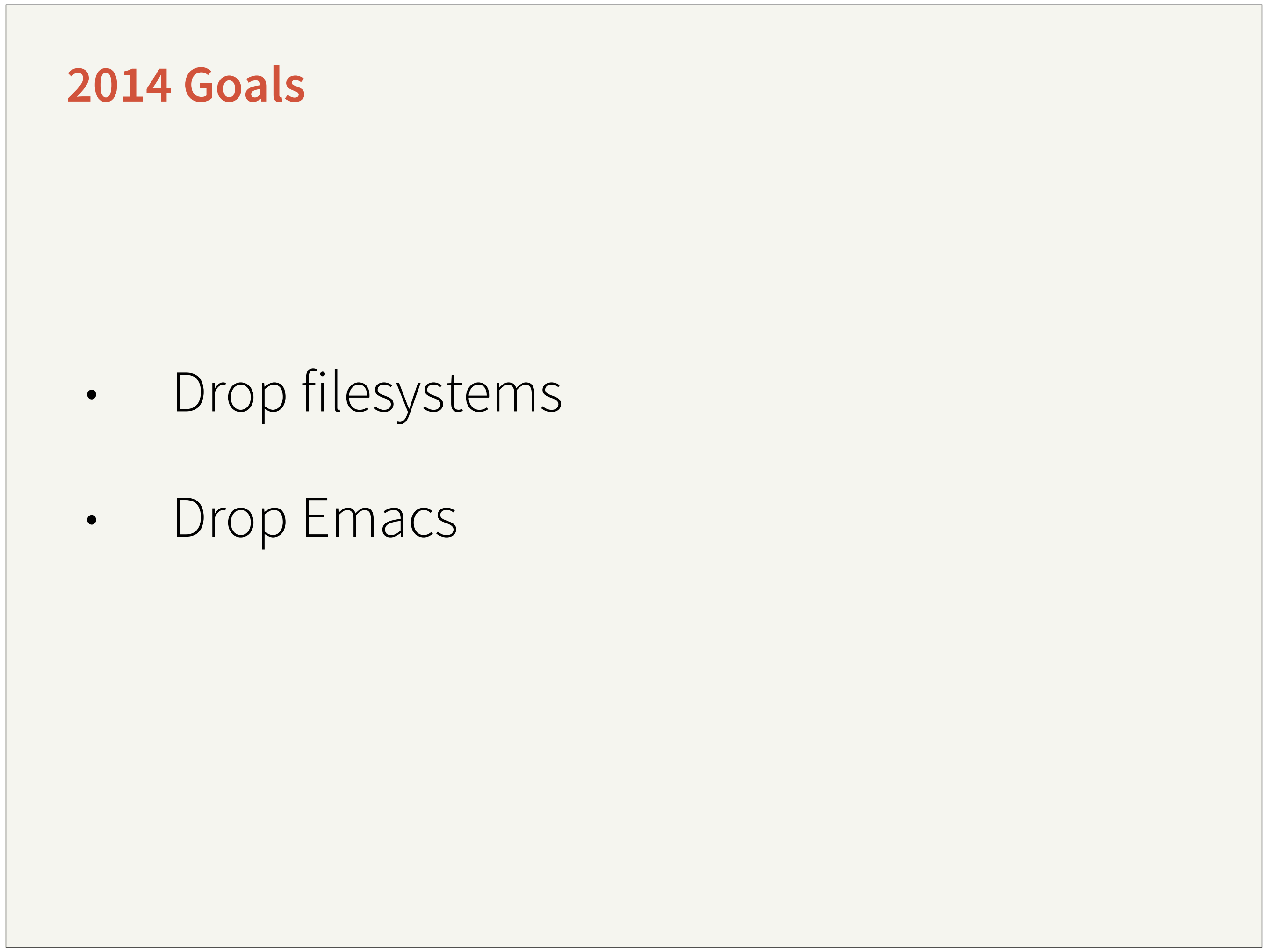
Tooling complexity (i.e., programs taking only text input) is baseline

Text sidesteps model v. view by only having one—text.

Datomic has simple tuple model; getting different views requires different queries.

Not clear how to choose which view to expose

clojure has ISeq, IReduce, IKVReduce; perf-oriented protocols vs. semantic



2014 Goals

- Drop filesystems
- Drop Emacs

Already have chunked content store design + prototype

Emacs is tricker—dropping it will require search; structural code editing; git interface

Unifying interfaces

Kevin Lynagh
SpeakerConf Barcelona
May 2014