pfil, firewalls and locking

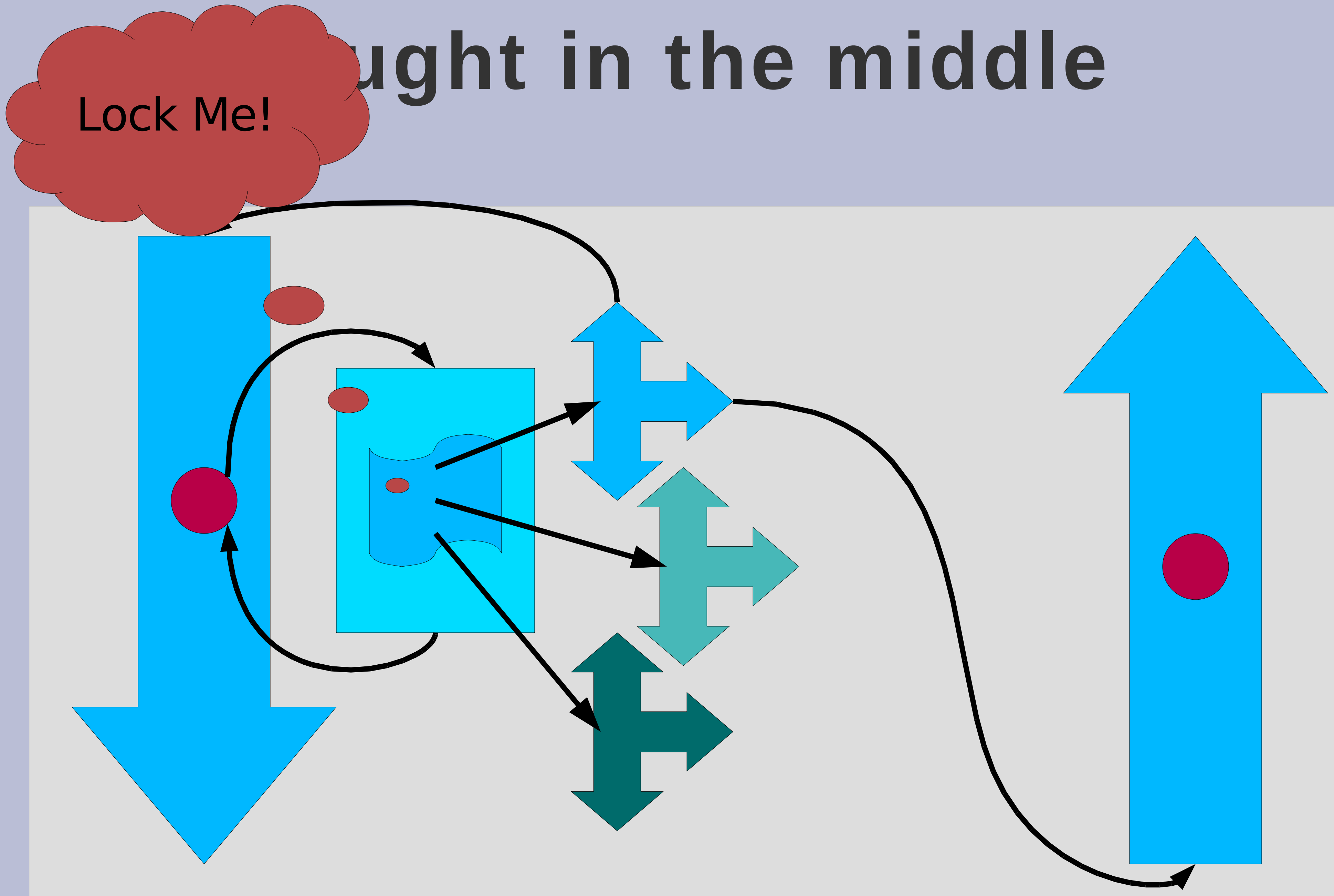
- Pfil(9) provides general hook points for packet filtering
- Current users: ipfw, pf and ipf
- Hooks in: ip[6]_{in,out}put
- Possibly also for L2 use

Pro/ Con

- Keeps the code clean
- Allows (almost) seamless integration of 3rd Party packet filters
- Allows to run more than one packet filter (and people do do that)
- Very good for developing and testing

- Overhead
- Locking (next slide)
- Changes in the main code can't be avoided completely, anyway

ught in the middle



Current approach

- rwlock(9) protected TAILQs
  + Allows for the recursion
  + Does not kill concurrency possibilities
  - Still needs atomic ops
  - Produces LORs (false positives?)
  - Writer starvation?

Layering (violations)

- Hook point is at the IP Layer
- User/group/jail rules look at the socket layer
- Entails LORs
- For the output path, this is solved: Pass the (locked) inpcb to the hooks
- For the input path: Inconclusive
  - Does a LOR between rw_rlock() and mtx_lock cause problems?

Plan?

1) Move to lockless/ static approach
   - Register hooks once and keep them
   - No atomic ops, no limitations for concurrency ... but no flexibility

2) Classic “read mostly” situation?
   - Gets rid of the atomic ops in the fast path ... but keeps the flexibility
   - Does not allow for recursion?