CAPSICUM

Practical capabilities for UNIX

19th USENIX Security Symposium
11 August 2010 - Washington, DC

Robert N. M. Watson — Google UK Ltd
Jonathan Anderson — FreeBSD Project
Ben Laurie — FreeBSD Project
Kris Kennaway — University of Cambridge

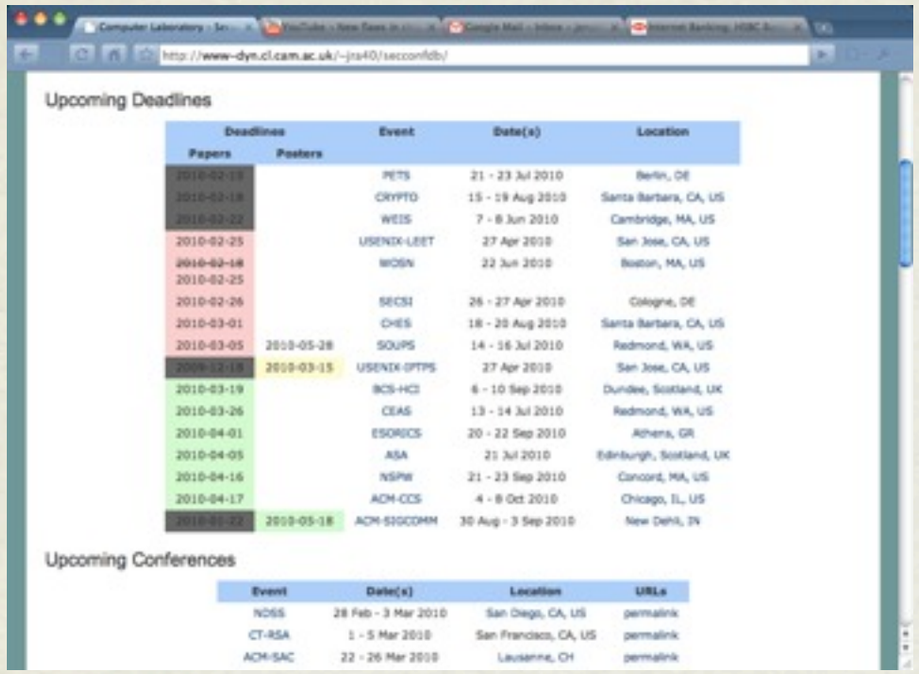
Introduction

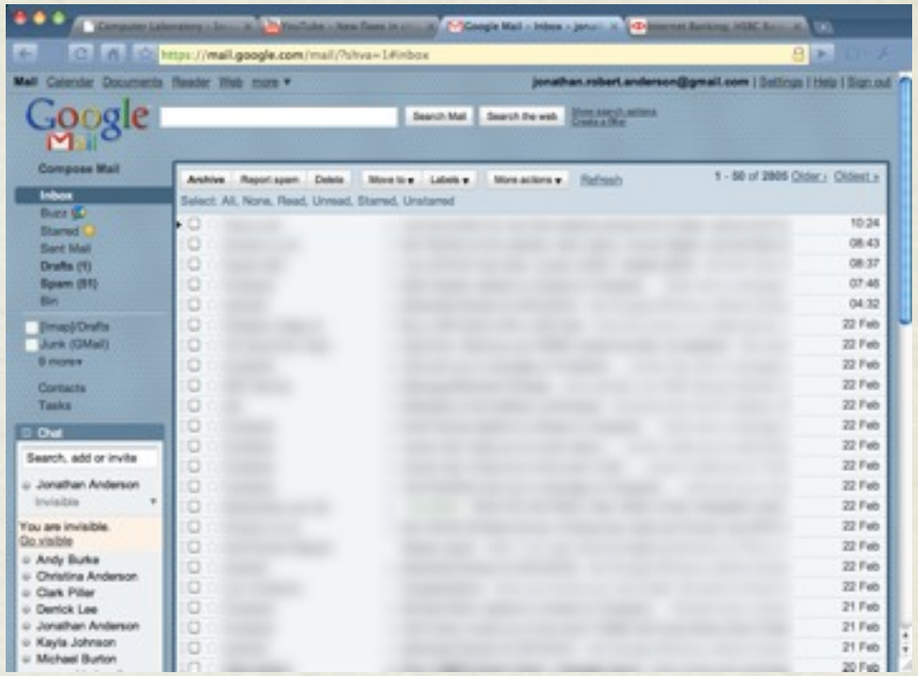
- Capsicum: hybrid UNIX/capability operating system
  - Requirements of complex, security-aware applications
  - Why MAC isn't quite what we want
  - Capsicum's *Capability Mode* and *Capabilities*
  - Interactions between applications and sandboxing
- Building on Capsicum

Paradigm shift
… change is ~~coming~~ here

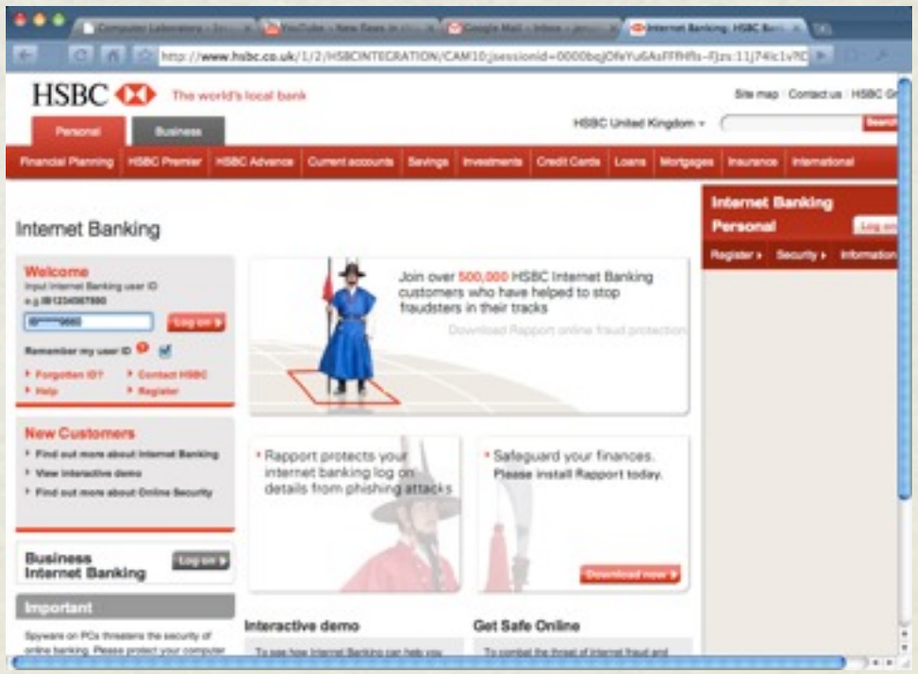
- Multi-user machines ➡ multi-machine users
- “Applications” frame competing interests
- Thin client one point of confluence

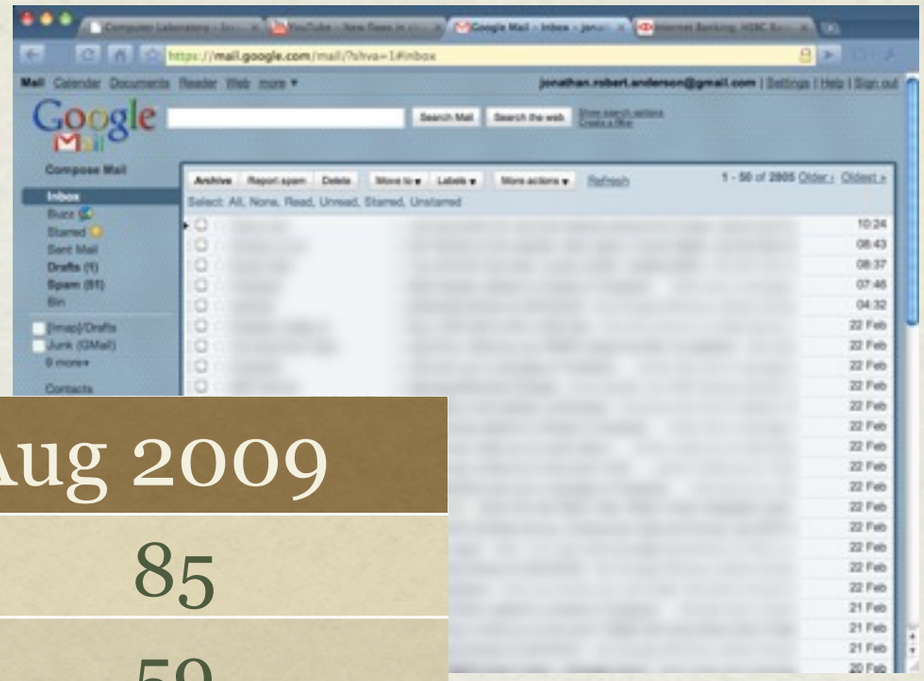
- DAC/MAC-centric access control ➡ sandboxing
- Application security rather than OS security
- Primitives for mapping distributed to local security domains


http://www-dyn.cl.cam.ac.uk/~jrs40/seccondb/
Upcoming Deadlines
Deadlines
Papers
Posters
Event
Date(s)
Location
2010-02-18
PETS
21 - 23 Jul 2010
Berlin, DE
2010-02-18
CRYPTO
15 - 19 Aug 2010
Santa Barbara, CA, US
2010-02-22
WEIS
7 - 8 Jun 2010
Cambridge, MA, US
2010-02-25
USENIX-LEET
27 Apr 2010
San Jose, CA, US
2010-02-18
2010-02-25
WOSN
22 Jun 2010
Boston, MA, US
2010-02-26
SECSI
26 - 27 Apr 2010
Cologne, DE
2010-03-01
CHES
18 - 20 Aug 2010
Santa Barbara, CA, US
2010-03-05
2010-05-28
SOUPS
14 - 16 Jul 2010
Redmond, WA, US
2009-12-18
2010-03-15
USENIX-OPTPS
27 Apr 2010
San Jose, CA, US
2010-03-19
BCS-HCI
6 - 10 Sep 2010
Dundee, Scotland, UK
2010-03-26
CEAS
13 - 14 Jul 2010
Redmond, WA, US
2010-04-01
ESORICS
20 - 22 Sep 2010
Athens, GR
2010-04-05
ASA
21 Jul 2010
Edinburgh, Scotland, UK
2010-04-16
NSPW
21 - 23 Sep 2010
Concord, MA, US
2010-04-17
ACM-CCS
4 - 8 Oct 2010
Chicago, IL, US
2010-01-22
2010-05-18
ACM-SIGCOMM
30 Aug - 3 Sep 2010
New Delhi, IN
Upcoming Conferences
Event
Date(s)
Location
URLs
NDSS
28 Feb - 3 Mar 2010
San Diego, CA, US
permalink
CT-RSA
1 - 5 Mar 2010
San Francisco, CA, US
permalink
ACM-SAC
22 - 26 Mar 2010
Lausanne, CH
permalink


https://mail.google.com/mail/?shva=1#inbox
Google Mail
Compose Mail
Inbox
Buzz
Starred
Sent Mail
Drafts (1)
Spam (91)
Bin
Contacts
Tasks
Chat
Search, add or invite
Jonathan Anderson
Invisible
You are invisible.
Go visible
Andy Burka
Christina Anderson
Clark Piller
Derrick Lee
Jonathan Anderson
Kayla Johnson
Michael Burton

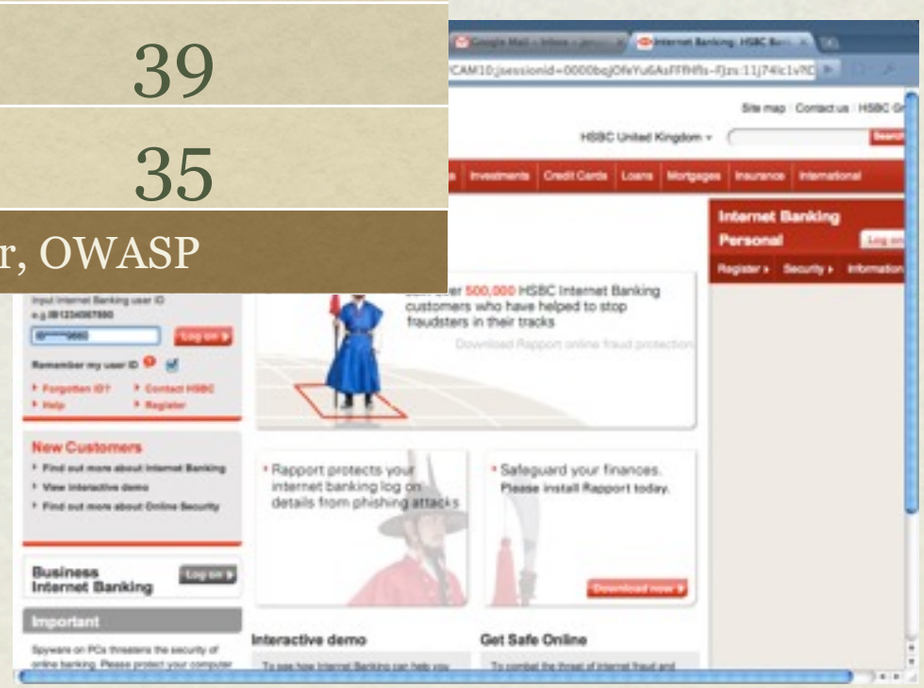

http://www.youtube.com/watch?v=_yyfcHSKZLc
YouTube
chip and pin site:youtube.com
Search
Create Account or Sign In
Home Videos Channels TV
Subscriptions History Upload
New flaws in chip and pin system revealed (BBC News) long version
Check the link in the sidebar for the full BBC news article
7 ratings
393 views
Favourite Share Playlists Flag
Facebook Bebo MySpace
Statistics & Data
More From: ChRiStIaAn008
Related Videos
New flaws in chip and PIN system revealed
UK chip and pin credit / debit cards are insecu...
Arab security system (funny)
How to Reverse-Engineer a Satellite TV Smart Card
How to Change Your T2 Pin


http://www.hsbc.co.uk/1/2/HSBCINTEGRATION/CAM10;jsessionid=0000bqjOfeYuGAsFFfHfls-fJzs:11j74ic1vf0
HSBC
The world's local bank
Personal
Business
HSBC United Kingdom
Financial Planning
HSBC Premier
HSBC Advance
Current accounts
Savings
Investments
Credit Cards
Loans
Mortgages
Insurance
International
Internet Banking
Welcome
Forgotten ID?
Contact HSBC
Help
Register
New Customers
Find out more about Internet Banking
View interactive demo
Find out more about Online Security
Business Internet Banking
Log on
Important
Join over 500,000 HSBC Internet Banking customers who have helped to stop fraudsters in their tracks
Rapport protects your internet banking log on details from phishing attacks
Safeguard your finances. Please install Rapport today.
Download now
Interactive demo
Get Safe Online



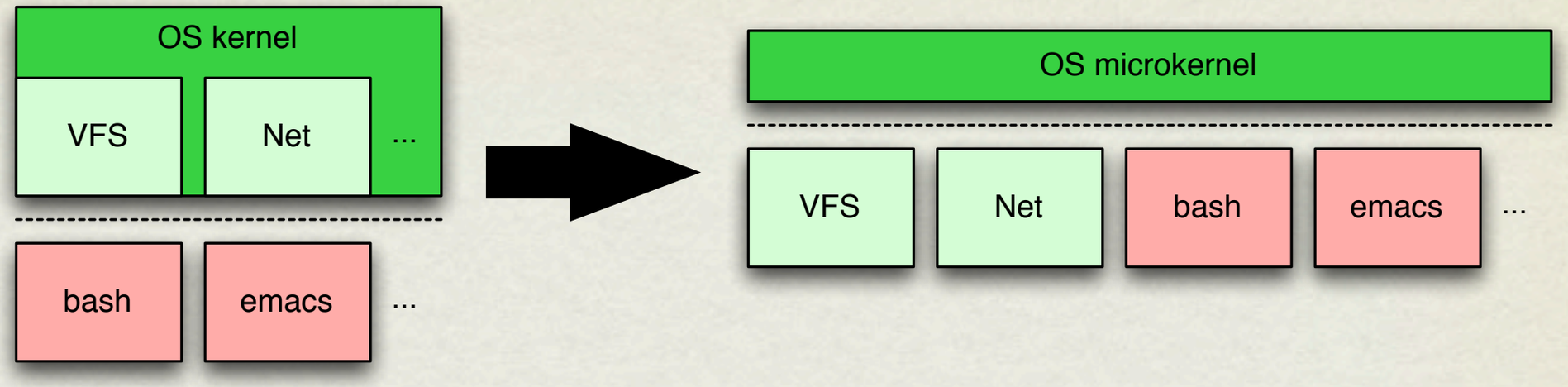
CVEs in Jan-Aug 2009	
Firefox	85
Safari	59
IE	48
Chrome	39
Flash	35

source; Justin Foster, OWASP



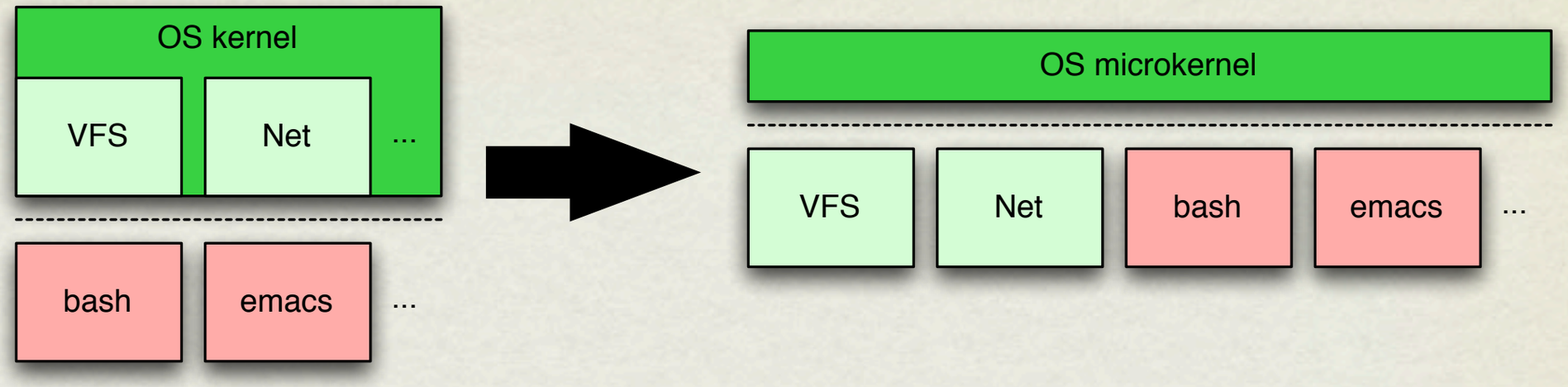
Microkernels to compartmentalisation

1980's

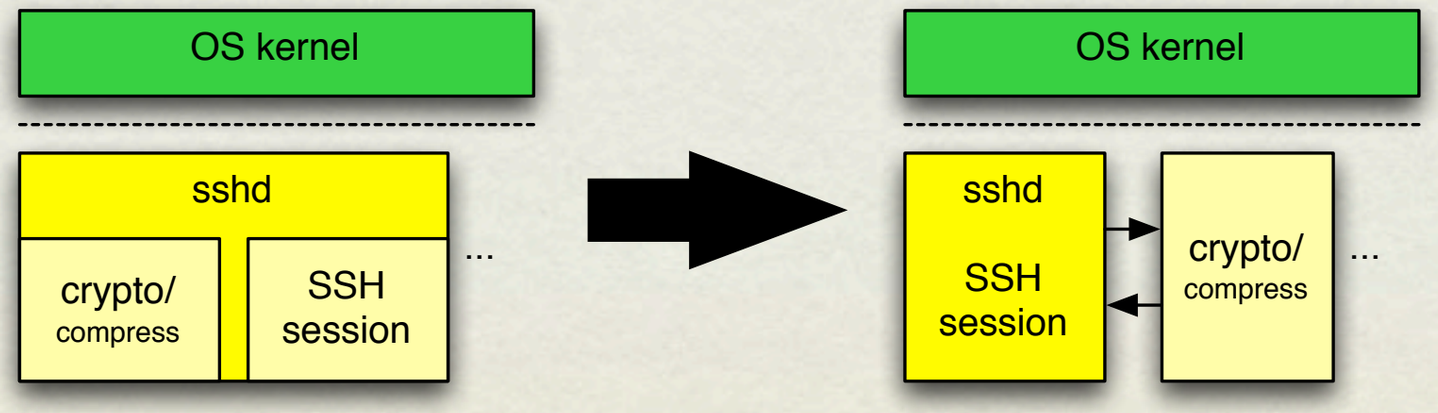


Microkernels to compartmentalisation

1980's



2000's



What about MAC?

	Type Enforcement (TE)	What we need
Interests of	Administrator	User or application
Sandbox creation	Administrator modifies global policy	On demand without using privilege
Policy source	Access control rules in global policy files	Embedded in applications, from UI

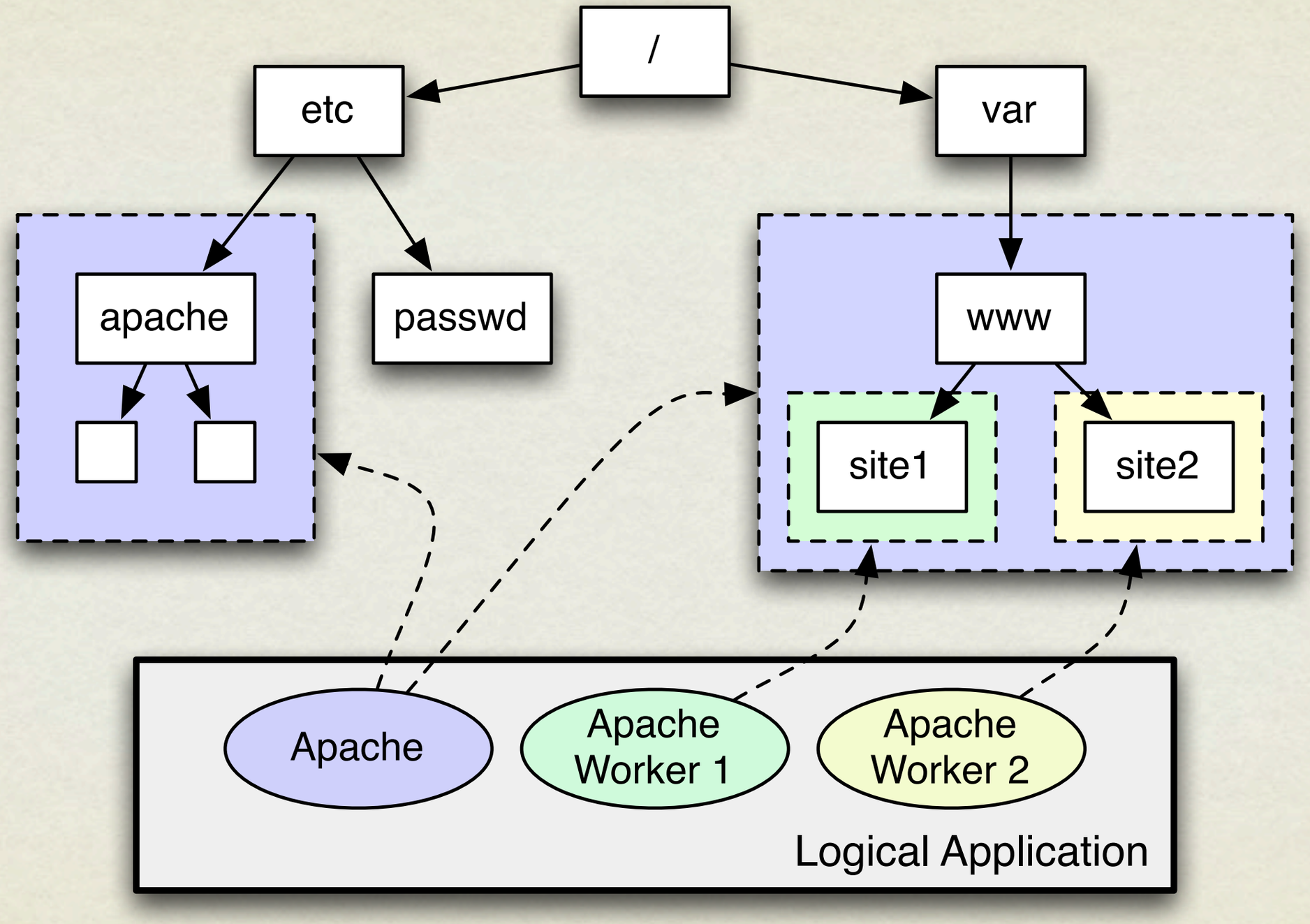
What about MAC?

	Type Enforcement (TE)	What we need
Interests of	Administrator	User or application
Sandbox creation	Administrator modifies global policy	On demand without using privilege
Policy source	Access control rules in global policy files	Embedded in applications, from UI

What about MAC?

	Type Enforcement (TE)	What we need
Interests of	Administrator	User or application
Sandbox creation	Administrator modifies global policy	On demand without using privilege
Policy source	Access control rules in global policy files	Embedded in applications, from UI

Application-driven rights delegation



Capability systems

A *capability* is an unforgeable token of authority.

Supports delegation-centric access control.

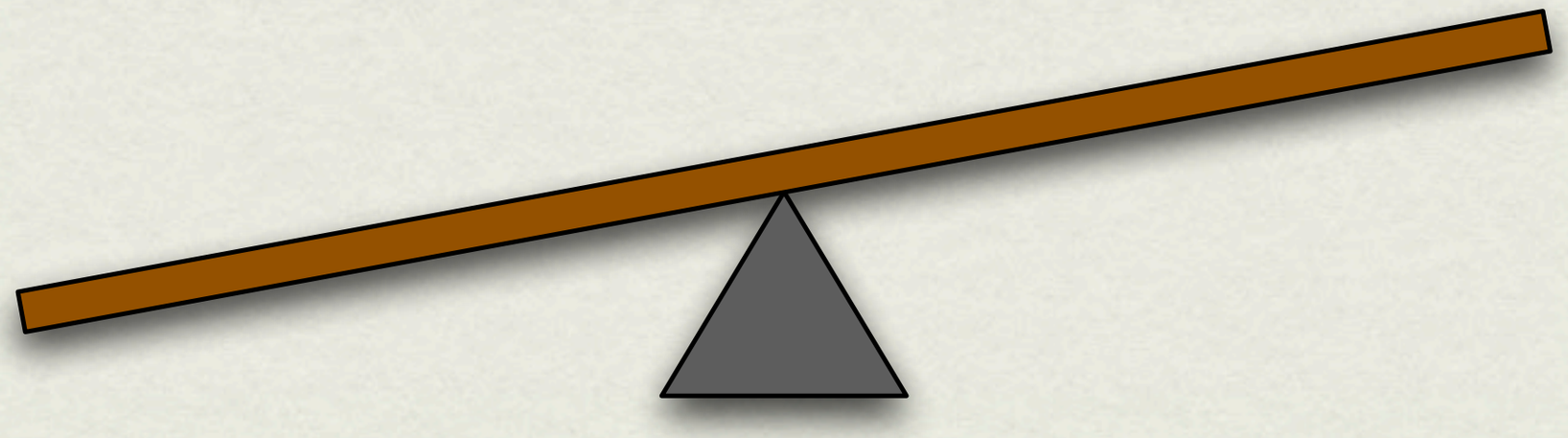
Where to start?

Production monolithic systems
UNIX, Linux, Windows, Mac OS X

✘ Monolithic kernel security model
✔ Real application stack today

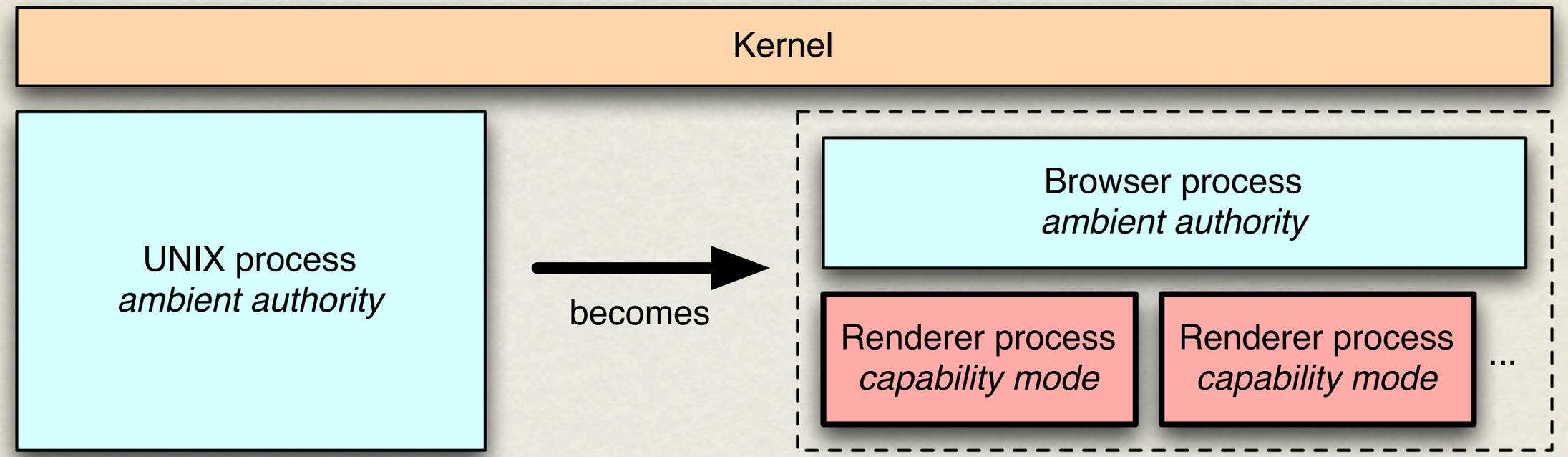
Research capability systems
EROS (CAPROS), CoyoteOS

✔ Least privilege design
✘ No extant application stack



Hybrid approach: **immediate security benefits** with a **long-term capability system vision**

Logical applications in Capsicum

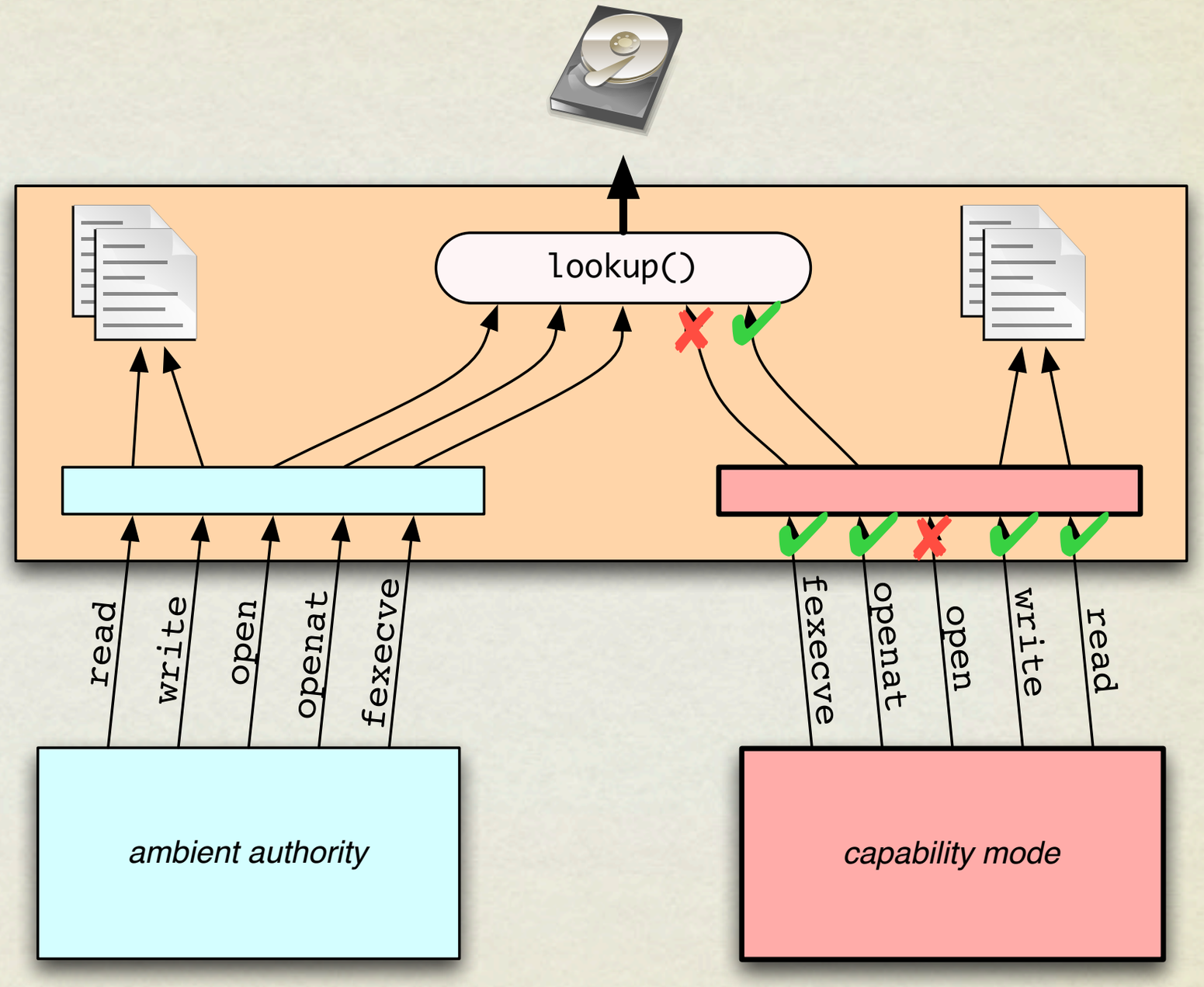


Traditional UNIX application

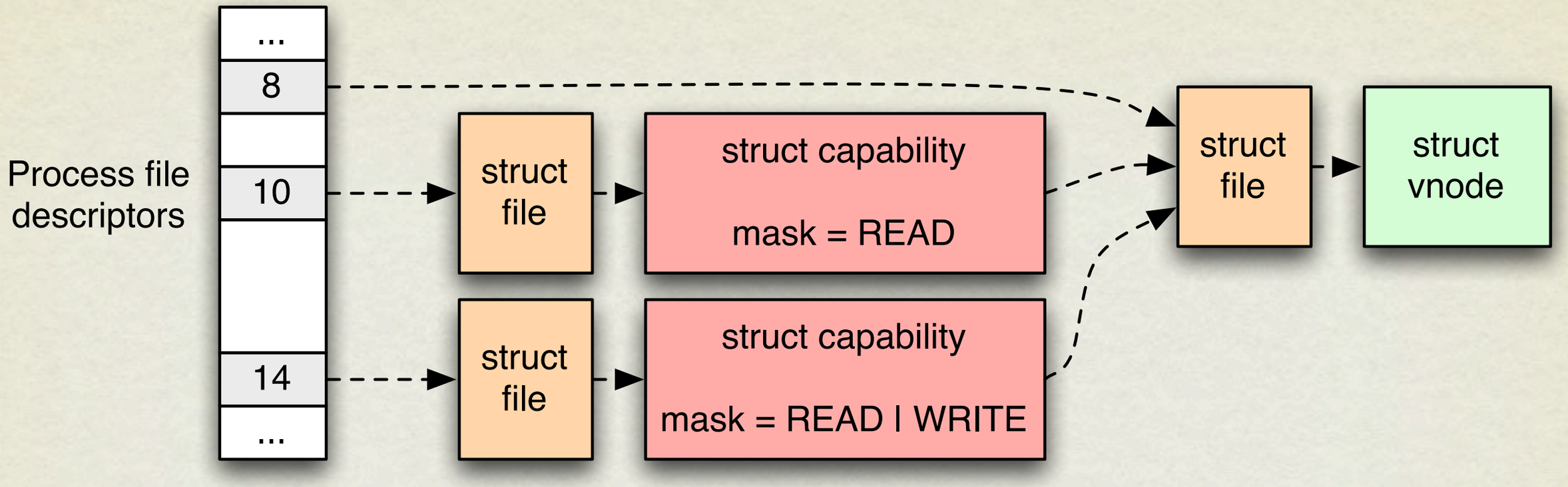
Capsicum logical application

Capability mode

- New system call `cap_enter` sets inherited credential flag
- Global OS name spaces restricted: only delegated rights available
- Interface thinning and other constraints on system calls

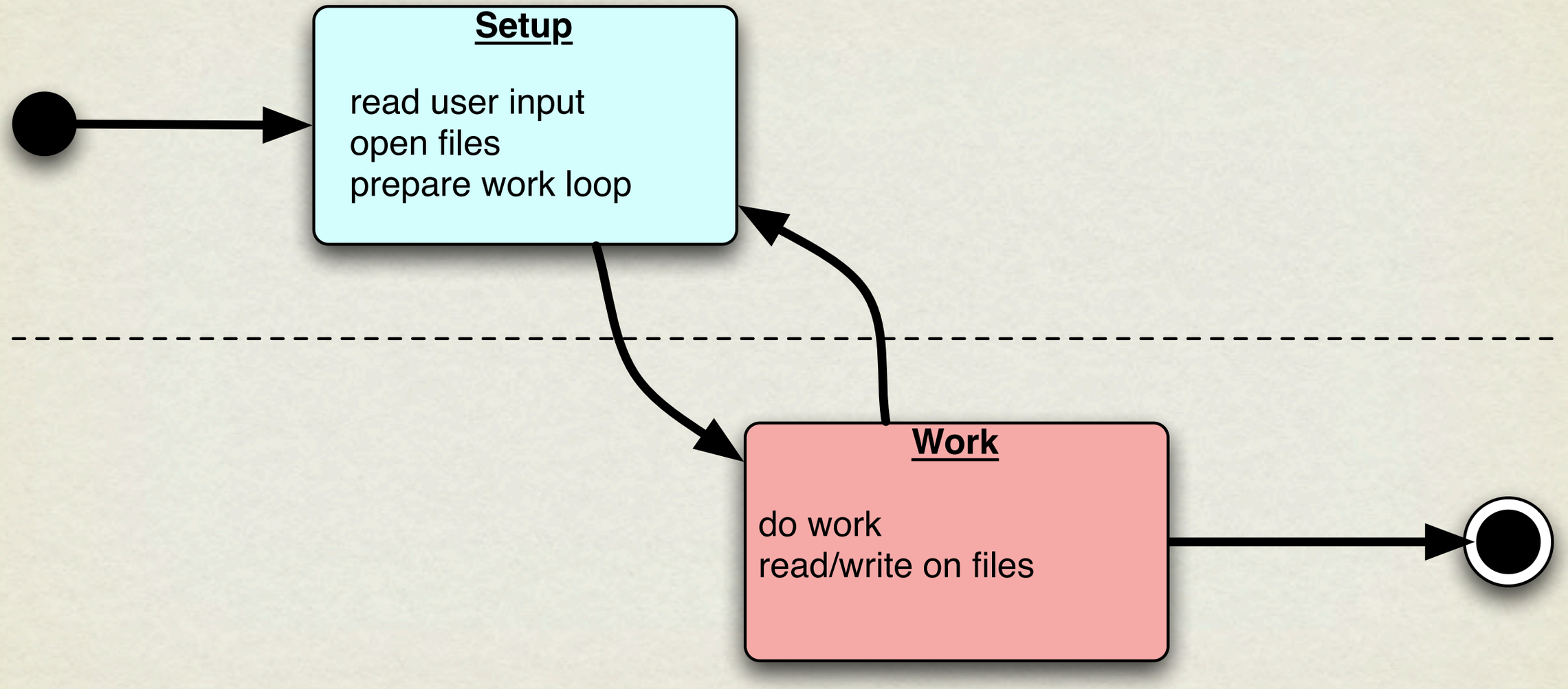


Capabilities

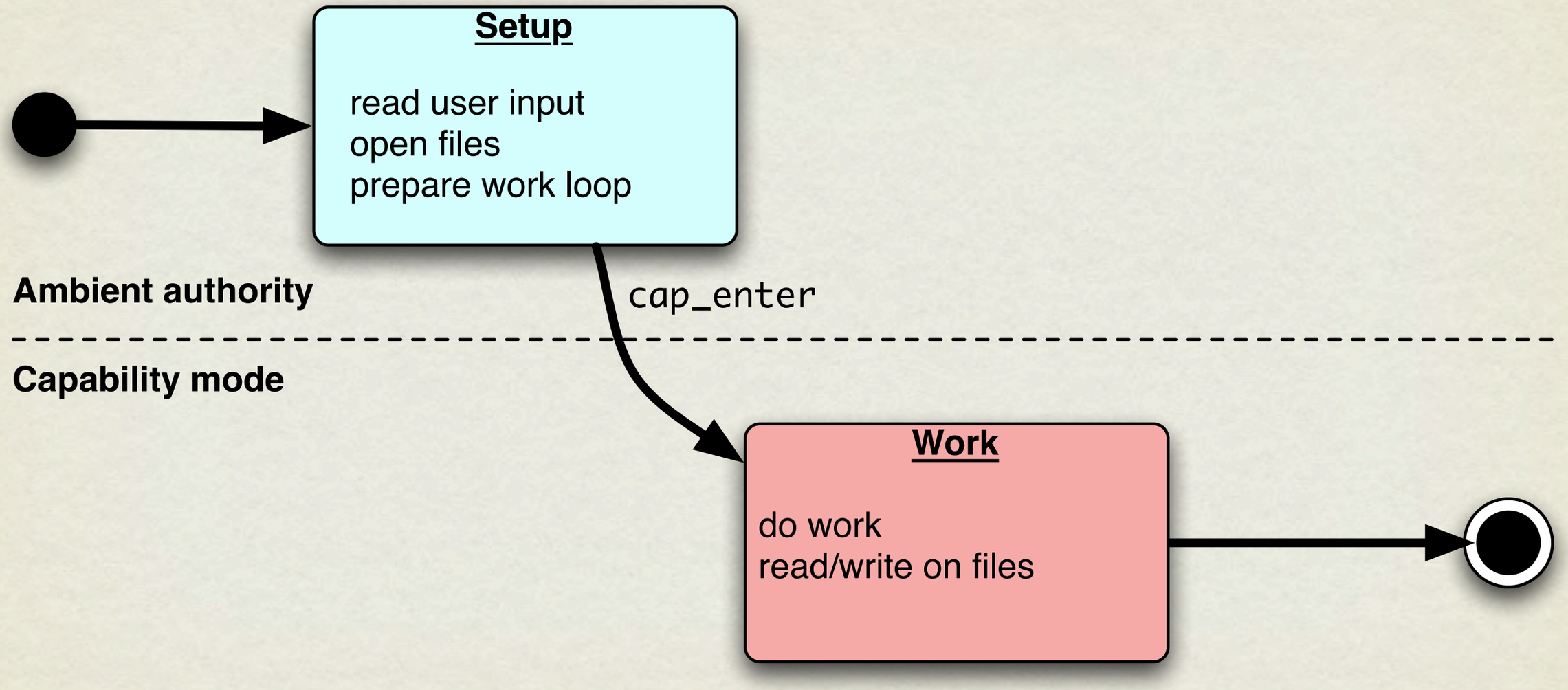


- Capabilities refine *open flags* on file descriptors
- `cap_new` on a capability further restricts access; no chains
- Inherited across `fork`/`exec` or passed via sockets
- Directory capabilities allow subtree delegation

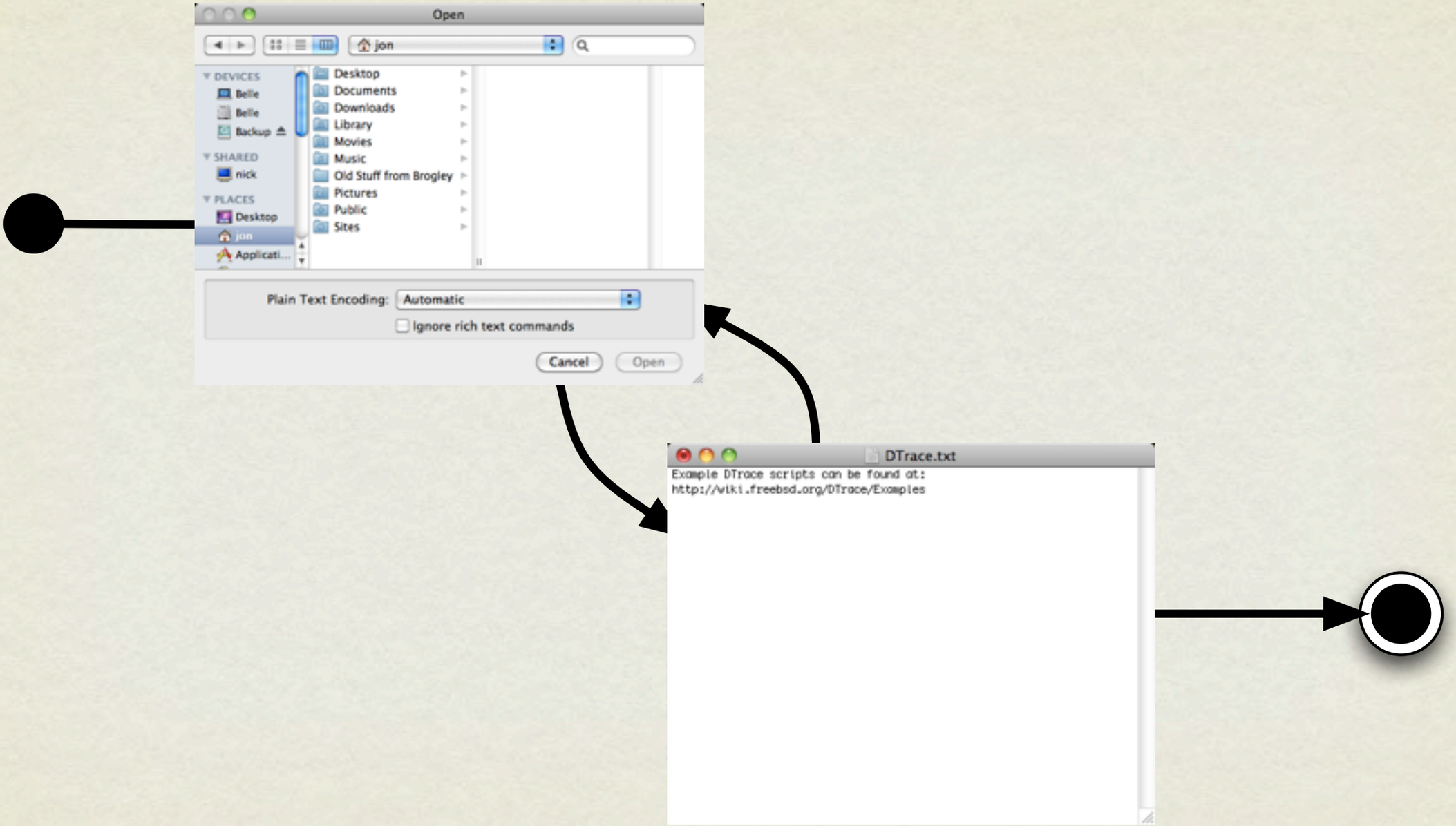
Possible application



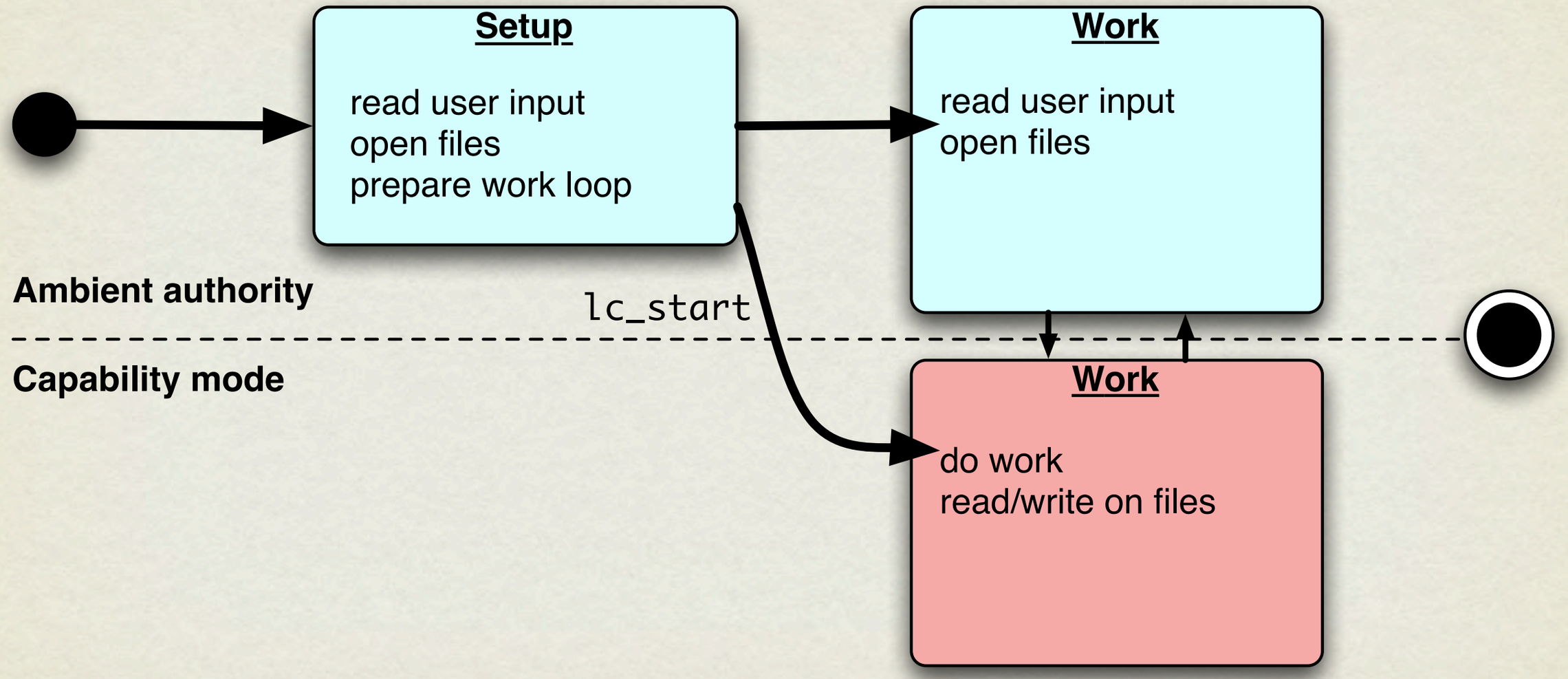
System call API



Interactive applications



libcapsicum API



Adapted applications

Program	Approach	Changes
tcpdump	`cap_enter`	Enter for parse/render work loop
dhclient	`cap_enter`	Reinforce existing `chroot`/`setuid` privilege separation
gzip	`libcapsicum`	Open files with ambient authority, pass capabilities to sandbox
Chromium	`cap_enter`	Sandbox Javascript and HTML processing in renderer processes

tcpdump

```
@@ -1197,6 +1199,14 @@
                (void)fflush(stderr);
        }
 #endif /* WIN32 */
+       if (lc_limitfd(STDIN_FILENO, CAP_FSTAT) < 0)
+               error("lc_limitfd: unable to limit STDIN_FILENO");
+       if (lc_limitfd(STDOUT_FILENO, CAP_FSTAT | CAP_SEEK | CAP_WRITE) < 0)
+               error("lc_limitfd: unable to limit STDIN_FILENO");
+       if (lc_limitfd(STDERR_FILENO, CAP_FSTAT | CAP_SEEK | CAP_WRITE) < 0)
+               error("lc_limitfd: unable to limit STDERR_FILENO");
+       if (cap_enter() < 0)
+               error("cap_enter: %s", pcap_strerror(errno));
        status = pcap_loop(pd, cnt, callback, pcap_userdata);
        if (WFileName == NULL) {
```

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	22,350	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	11,300	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	**22,350**	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	**11,300**	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	22,350	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	11,300	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	22,350	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	11,300	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	22,350	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	11,300	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Chromium sandboxing

	OS	Sandbox	LoC	FS	IPC	NET	S≠S′	Priv
DAC	Windows	DAC ACLs	22,350	⚠	⚠	✘	✘	✔
	Linux	chroot()	600	✔	✘	✘	✔	✘
MAC	Mac OS X	Sandbox	560	✔	⚠	✔	✔	✔
	Linux	SELinux	200	✔	⚠	✔	✘	✘
Cap	Linux	seccomp	11,300	⚠	✔	✔	✔	✔
	FreeBSD	Capsicum	100	✔	✔	✔	✔	✔

Building on Capsicum

- Assisted compartmentalisation (static, dynamic analysis)
- Critical network services: routing daemon, etc.
- Monolithic applications: OpenOffice.org, KDE..
- Distributed domains ➡ local domains: browsers, databases…
- Gesture-Based Access Control (GBAC)
  - Power boxes, “Drag and drop” ➡ assign capabilities

Conclusion

- Multi-user security ➡ compartmentalised applications
- Capsicum APIs **faster**, **cleaner**, and **more secure**
  - Delegation-centric approach to granular policy
  - Avoid policy dual-coding, no privilege requirement
- **Supplement** rather than replace DAC and MAC
- API/semantics + prototype on FreeBSD 9.x, 8.x backport
- Linux/ChromeOS port in progress at Google

Questions?

* http://www.cl.cam.ac.uk/research/security/capsicum/