Forensic Jailbreaking of iOS devices

Dr. Bradley Schatz

Director, Schatz Forensic

V1.0 DFRWS USA

About me

- Dr Bradley Schatz
  - PhD, Digital Forensics (2007) ; BSc, Computer Science
- Schatz Forensic / Evimetry (2009-)
  - Practitioner, R&D, tool vendor
- Research affiliations
  - DFRWS Conference USA, Chair (2019), Technical Program Committee Chair (2017)
  - Journal of Digital Investigation (Editorial Board)
- Practical contributions
  - Volatility Memory Forensics Framework (Vista & Windows 7 support) (2010)
  - AFF4
  - Autopsy (index.dat support)
- Queensland University of Technology
  - Adjunct associate professor, doctoral supervision

iOS acquisition completeness is dwindling

*For private practice examiners

- Current backup-based logical imaging
  - No email,
  - No SQLite write ahead logs
  - Large swaths of filesystem and useful traces missing

- * CAIS/Greykey
  - Will produce complete logical images for govt. licencees
  - Will they assist in Civil matters? Not in my experience.

Exploitation/Jailbreaking is increasingly being used in civil forensic practice

- Forensic questions
  - Was my phone compromised?
  - Can I get deleted text messages?
  - What time was a voice message first recorded?
  - Deleted data recovery (SQLite WAL)
  - Inaccessible information

Current approaches in a nutshell

- Download jailbreak from internet
- Install and run jailbreak on the suspect iPhone*
- Install SSHD using Cydia
- Use SCP or netcat to copy the filesystem

* After you have tested it on a similar phone

iOS jailbreaking in forensics: literature

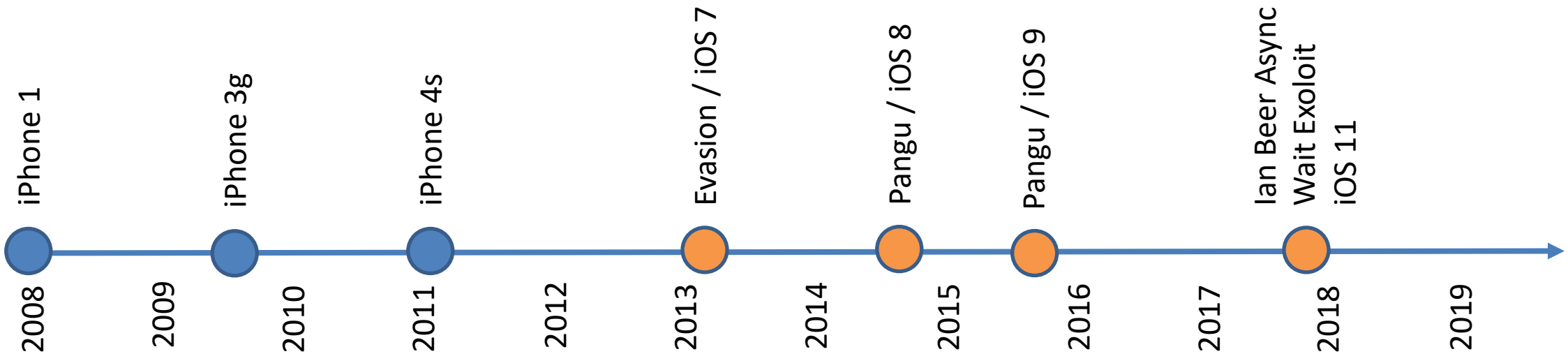
- Elcomsoft suggest the following jailbreaks to enable running their software
  - iOS 10:
    - h3lix (iOS 10.0-10.3.3), 32-bit devices, https://h3lix.tihmstar.net/
    - Meridian (iOS 10.0-10.3.3), 64-bit devices, https://meridian.sparkes.zone/
  - iOS 11:
    - LiberIOS (iOS 11.0-11.1.2), 64-bit devices, http://newosxbook.com/liberios/
    - Electra (iOS 11.0-11.1.2), 64-bit devices, https://coolstar.org/electra/

- Sara Edwards* suggests the following, with an open source methodology
  - iOS 11:
    - LiberIOS (iOS 11.), 64-bit devices
    - Meridian (iOS 10), 64-bit devices

  * See “iOS imaging on the Cheap”

What is jailbreaking?

The evolution of iOS jailbreaking & forensics



Unsigned Recovery Ramdisk Boot

Public Bootrom Exploits

Private source jailbreaking

Open source jailbreaking

Zdzardiarsky

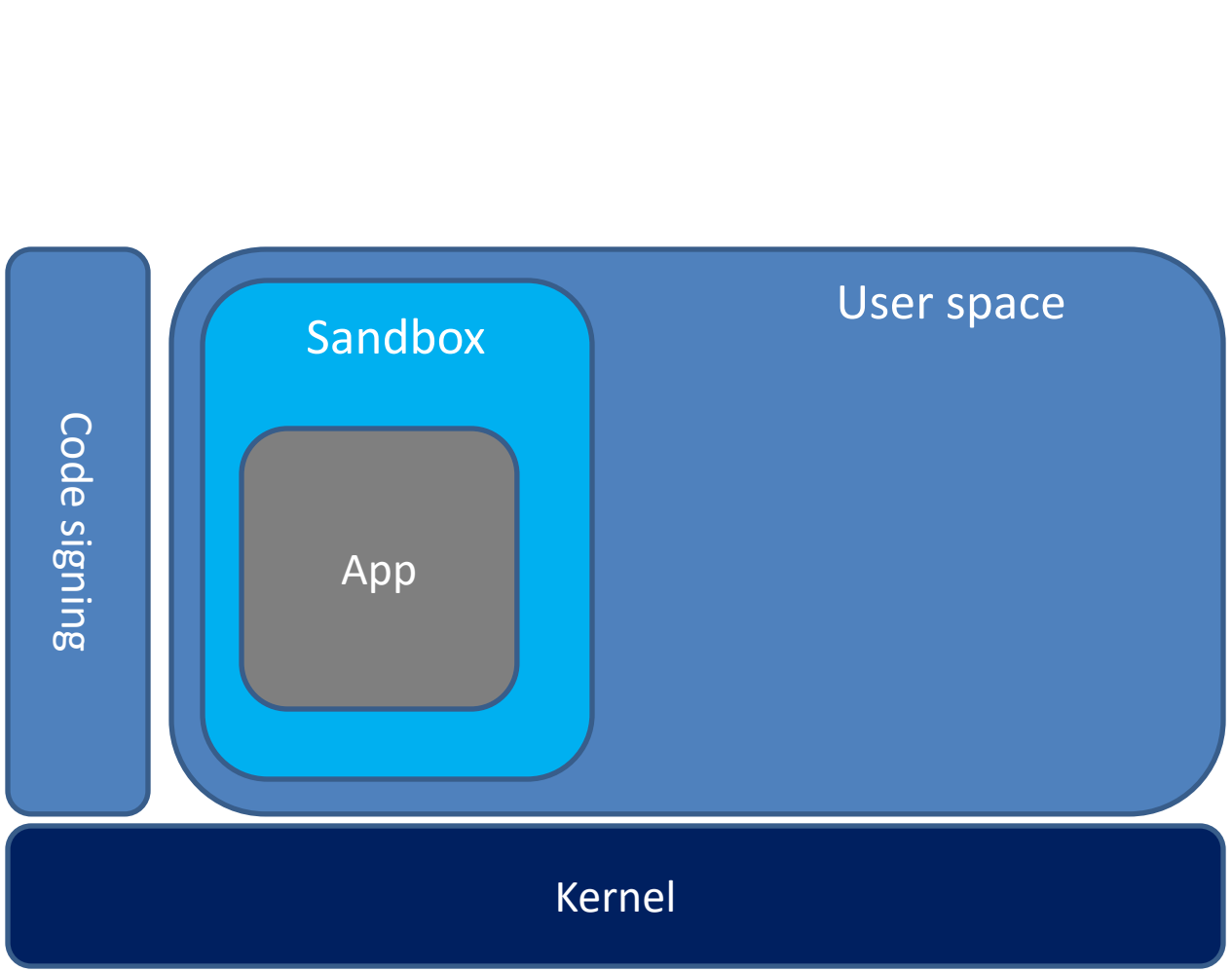
iPhone-dataprotection

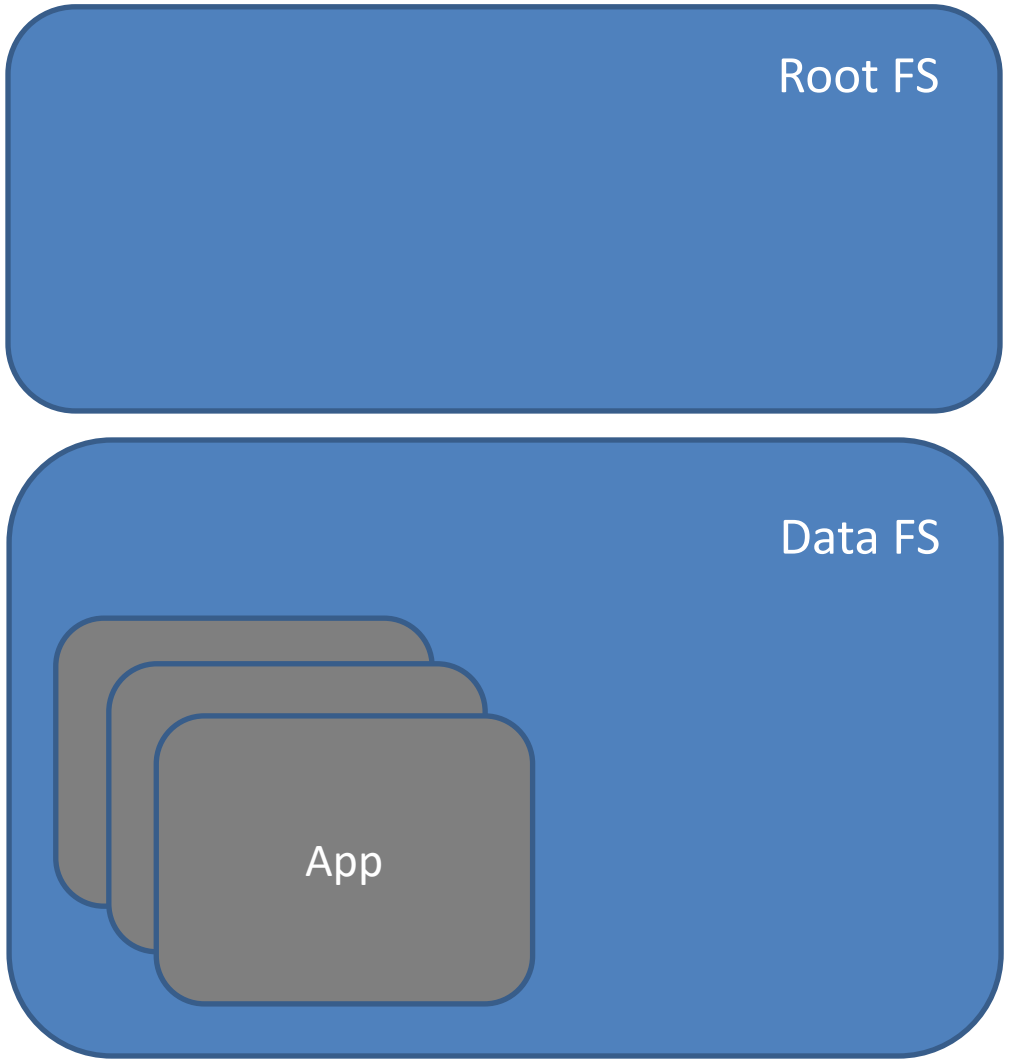
Greykey

CAIS?

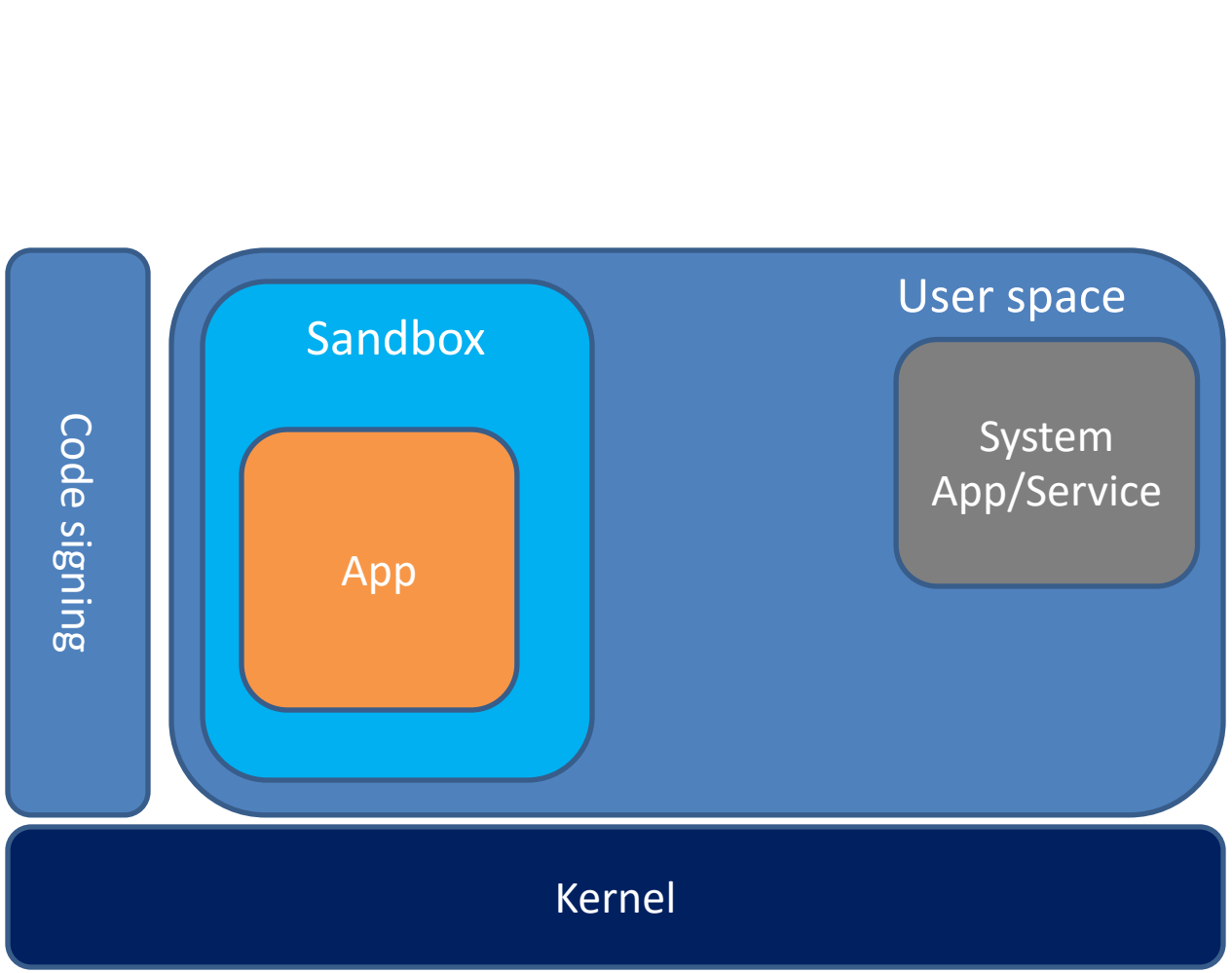
AFC/Backup

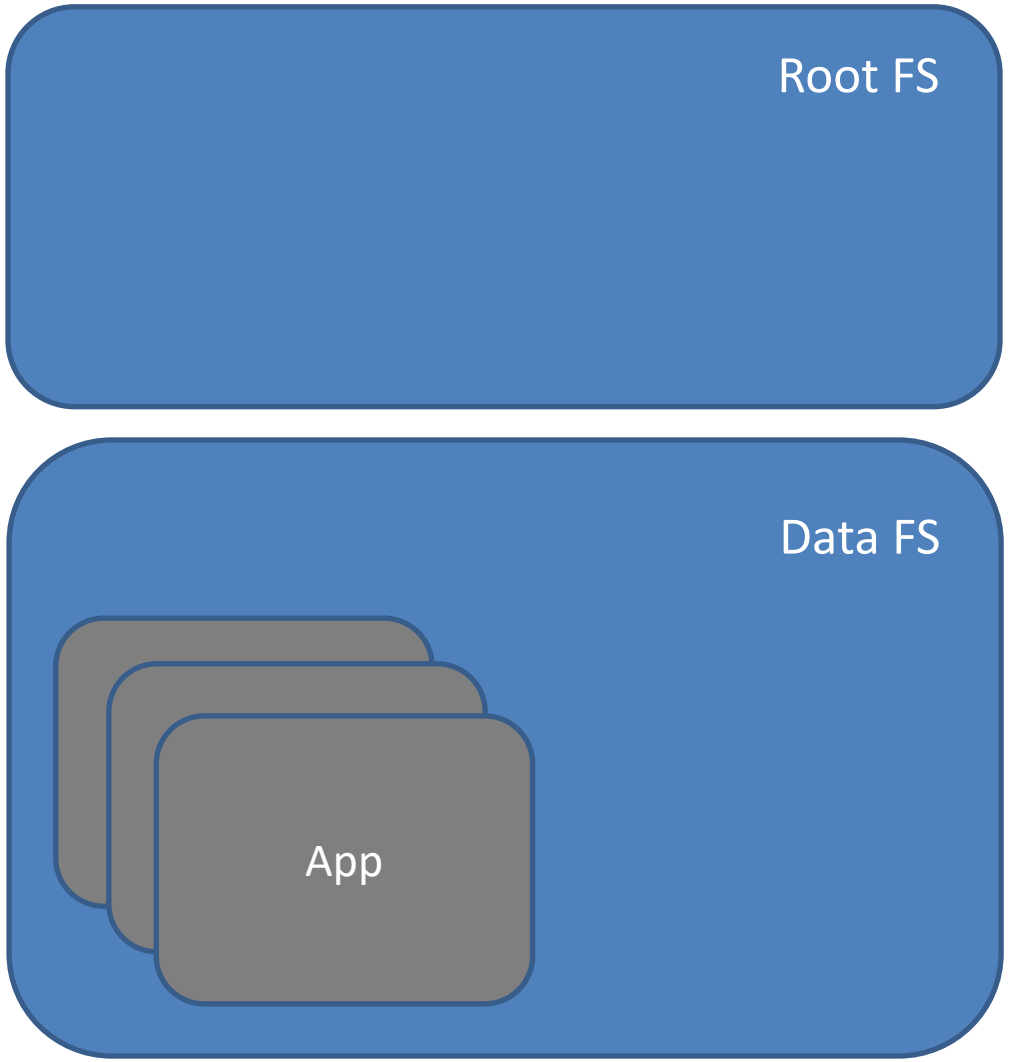
iOS security model (simplified)



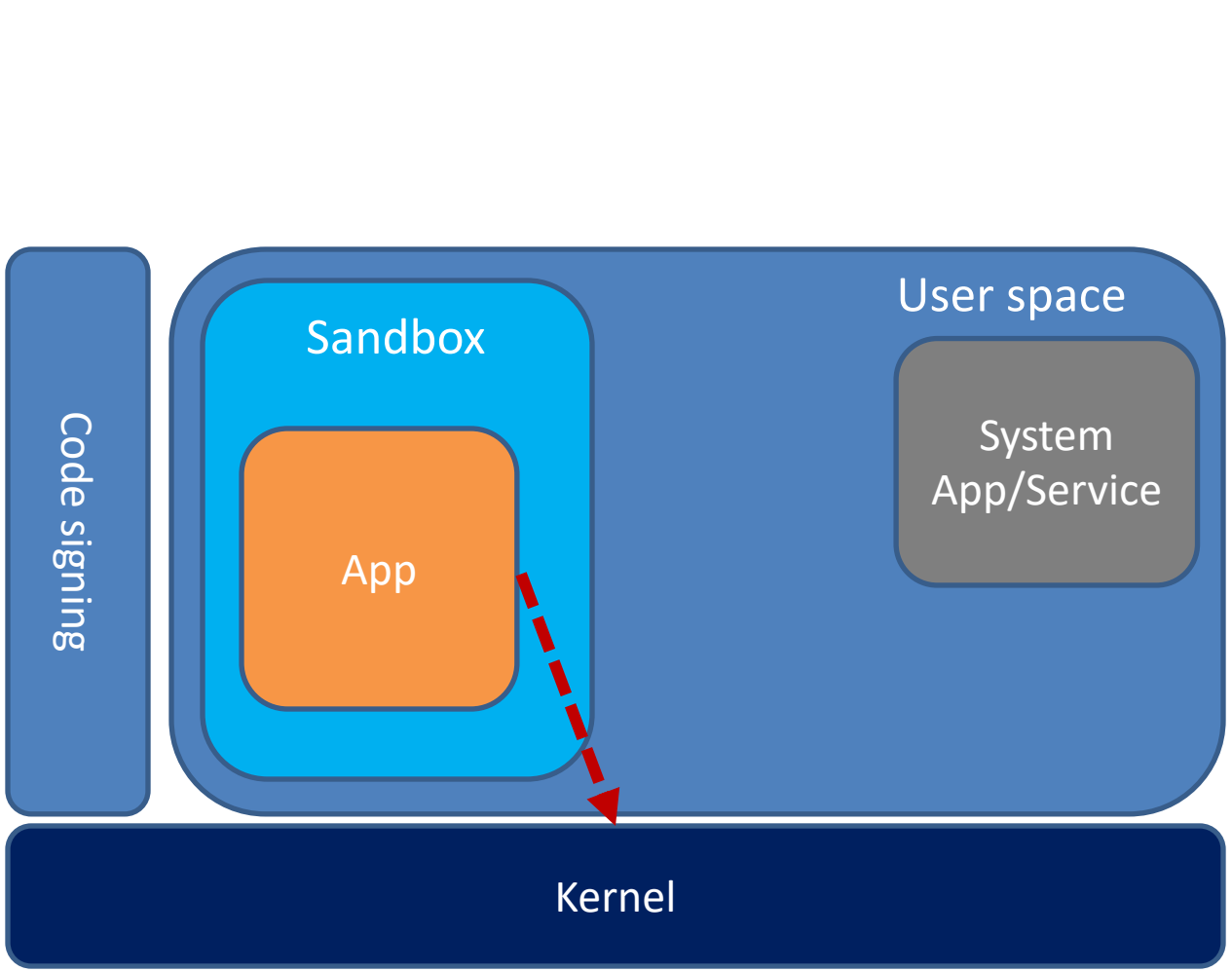


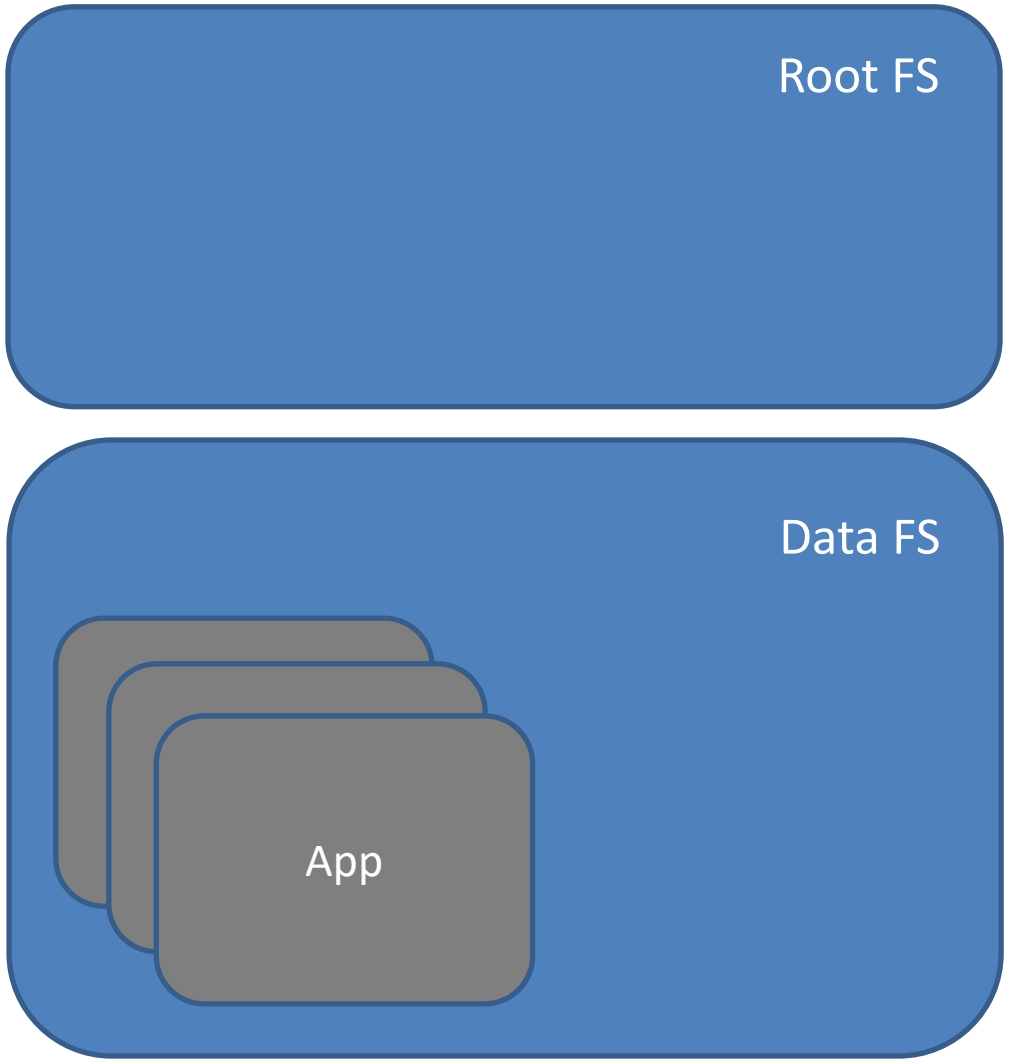
Anatomy of a JB: Code exec



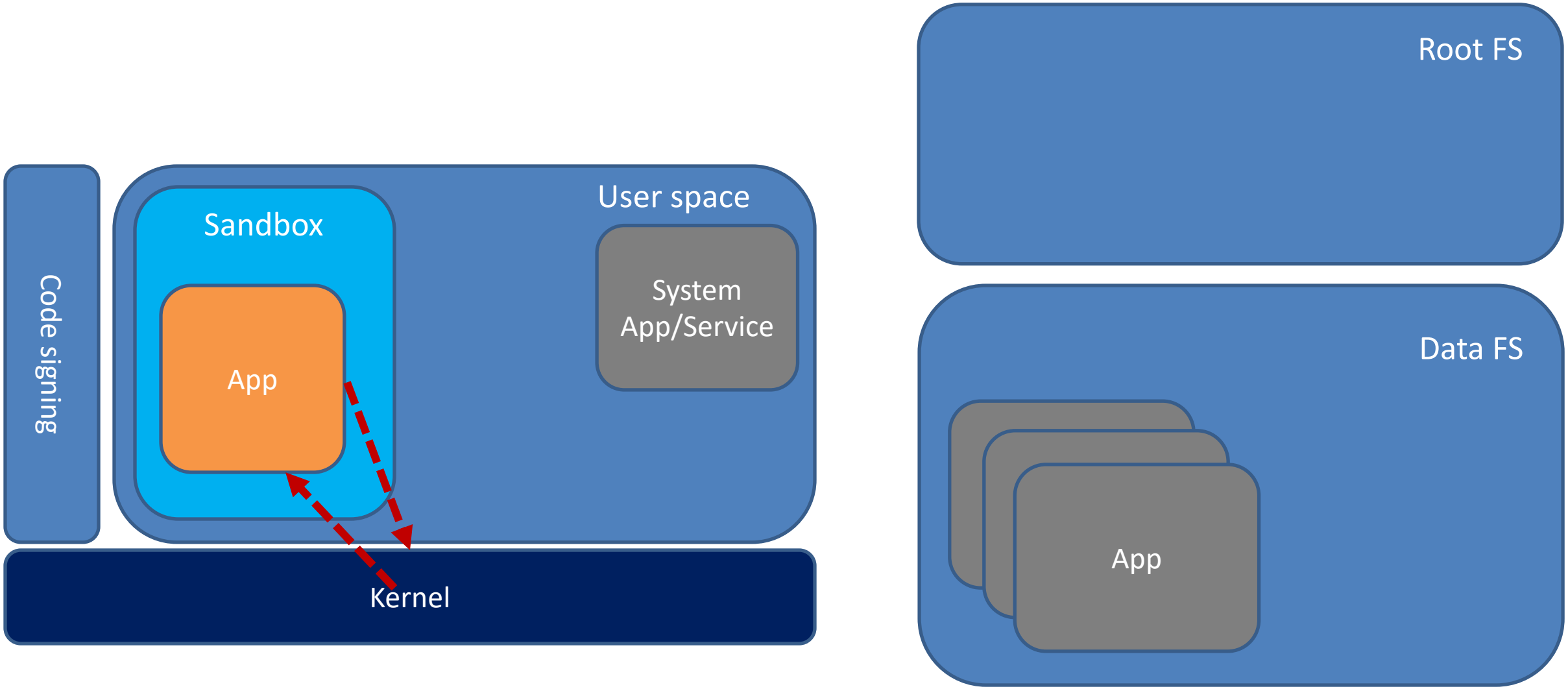


Anatomy of a JB: Sandbox escape/elevate privileges

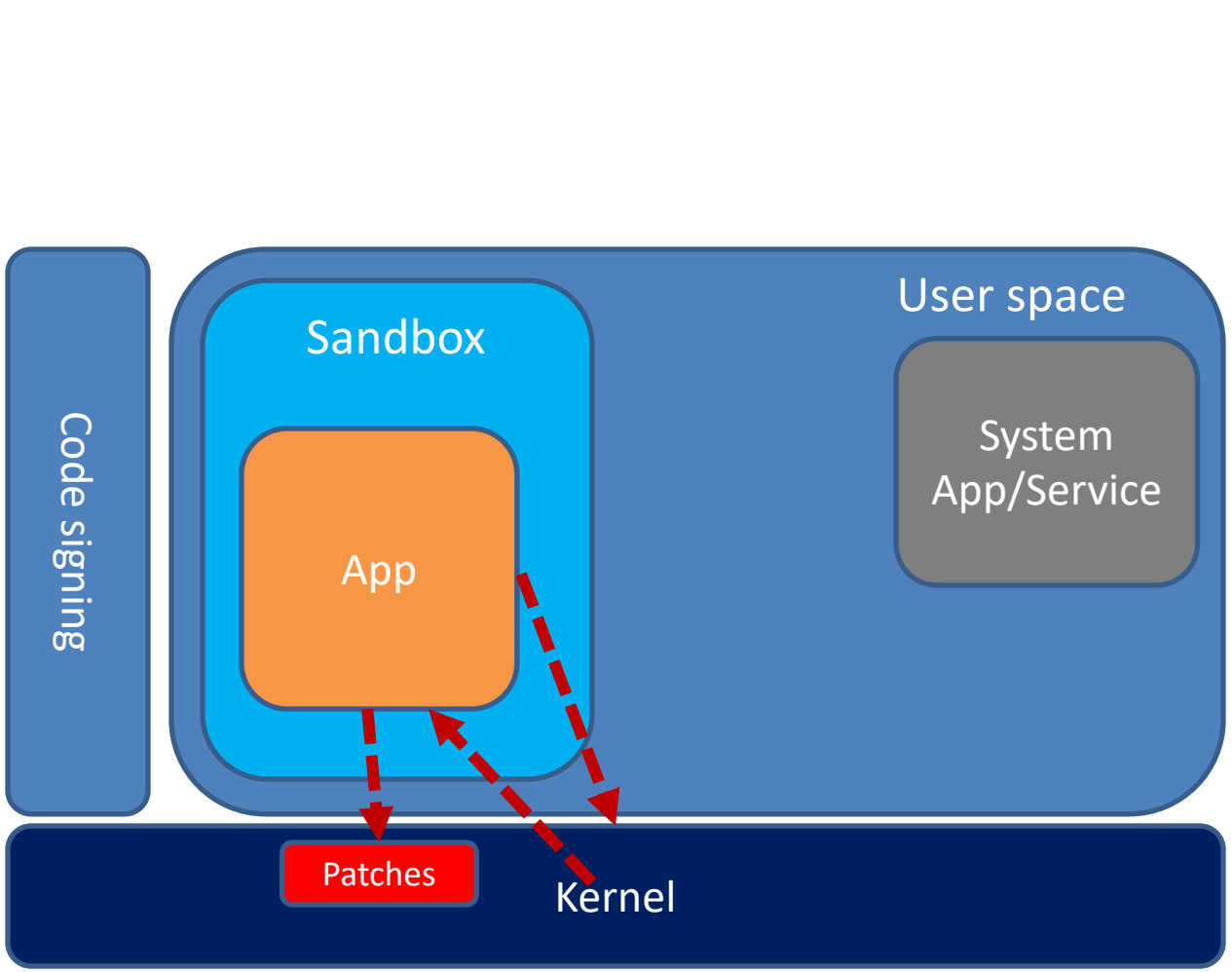


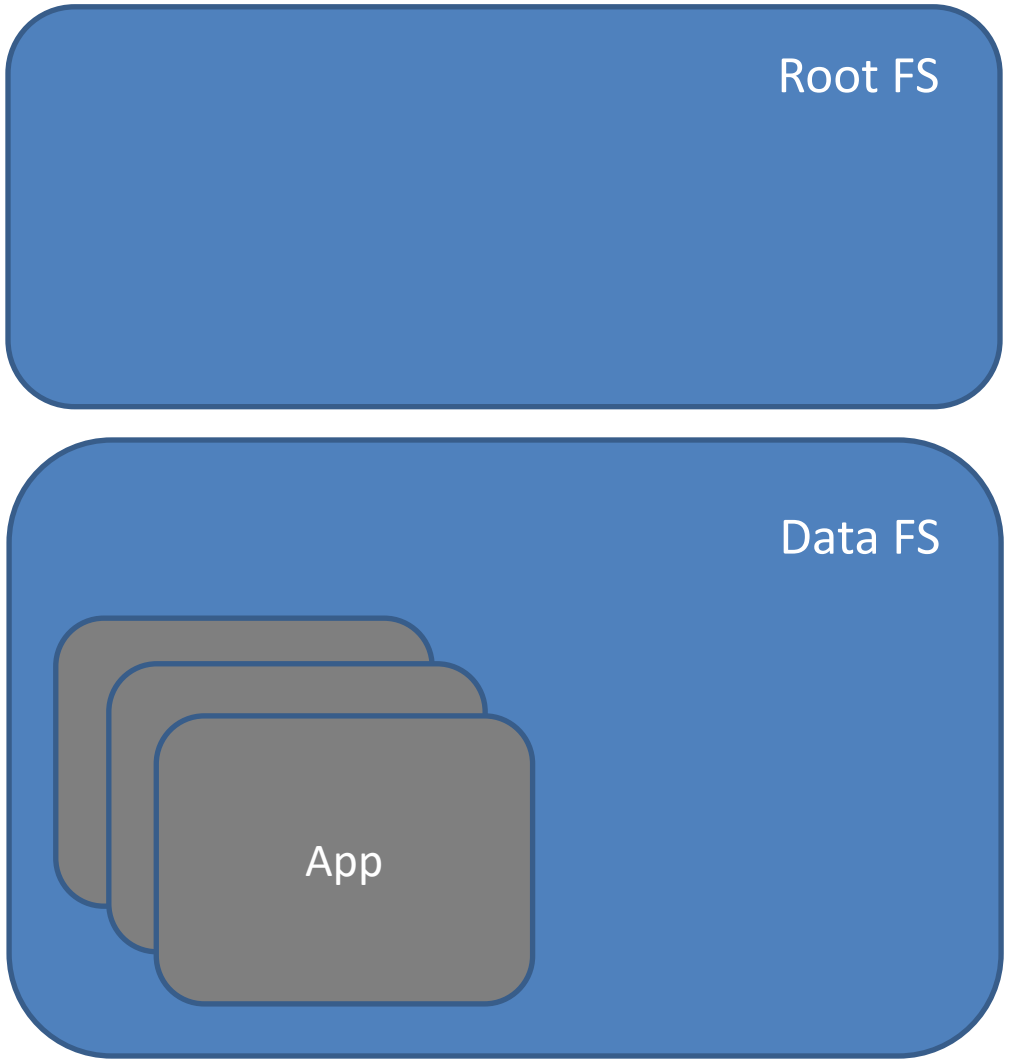


Anatomy of a JB: Read kernel

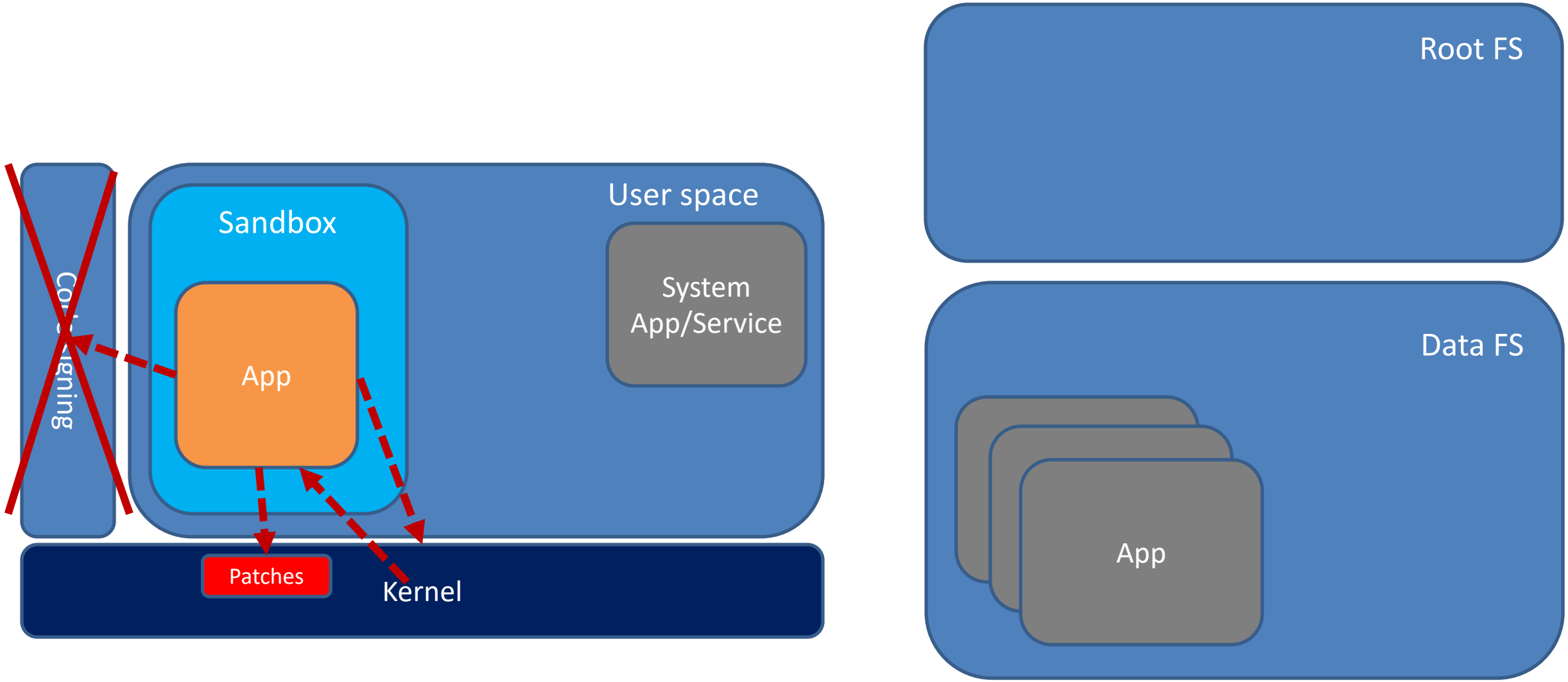


Anatomy of a JB: Patch kernel

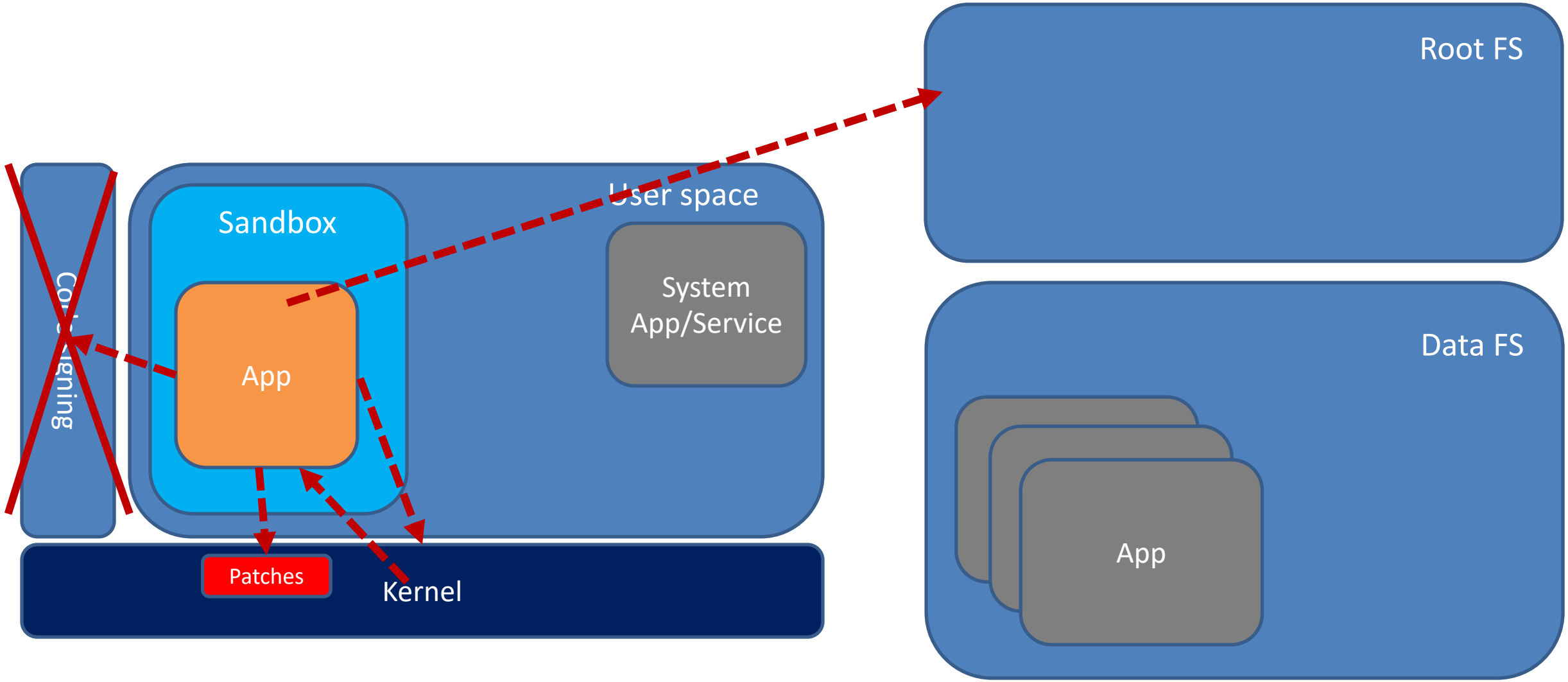




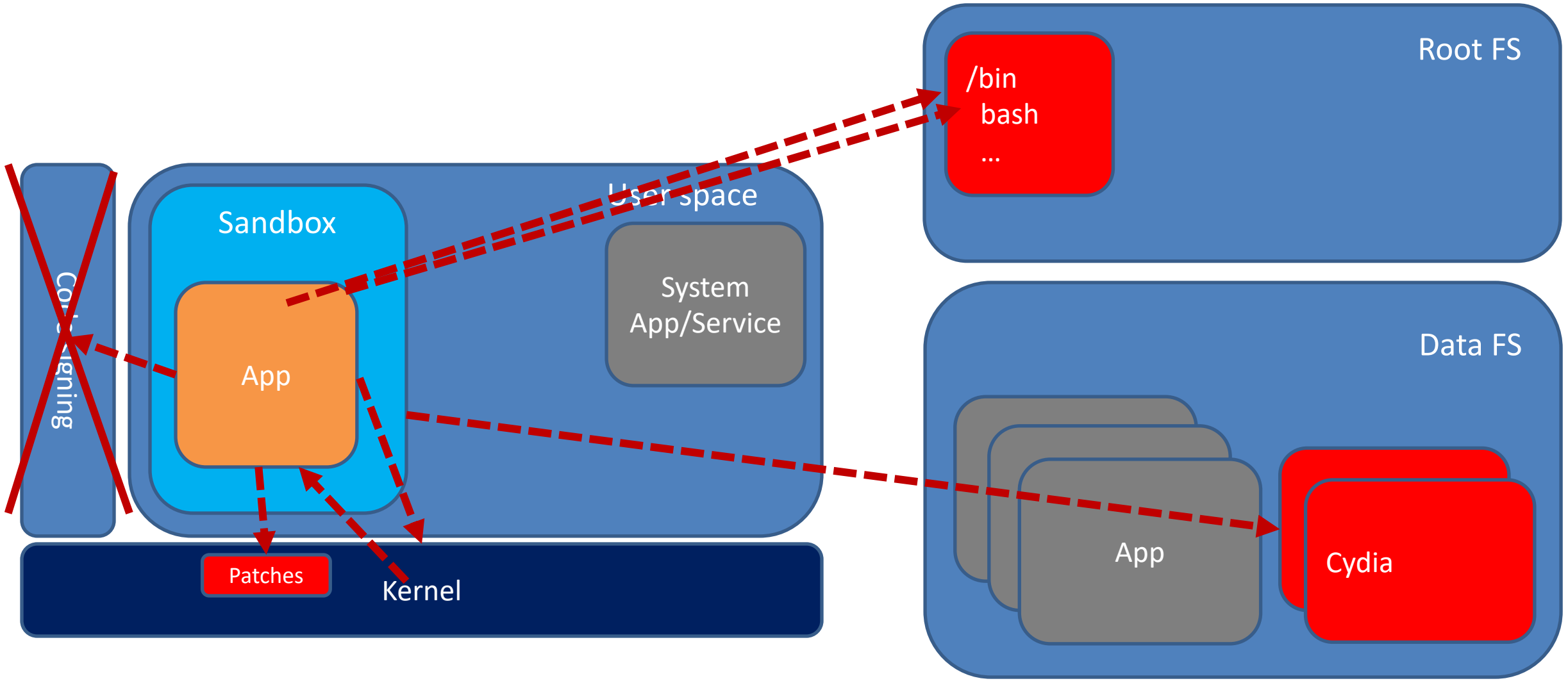
Anatomy of a JB: Neuter code signing



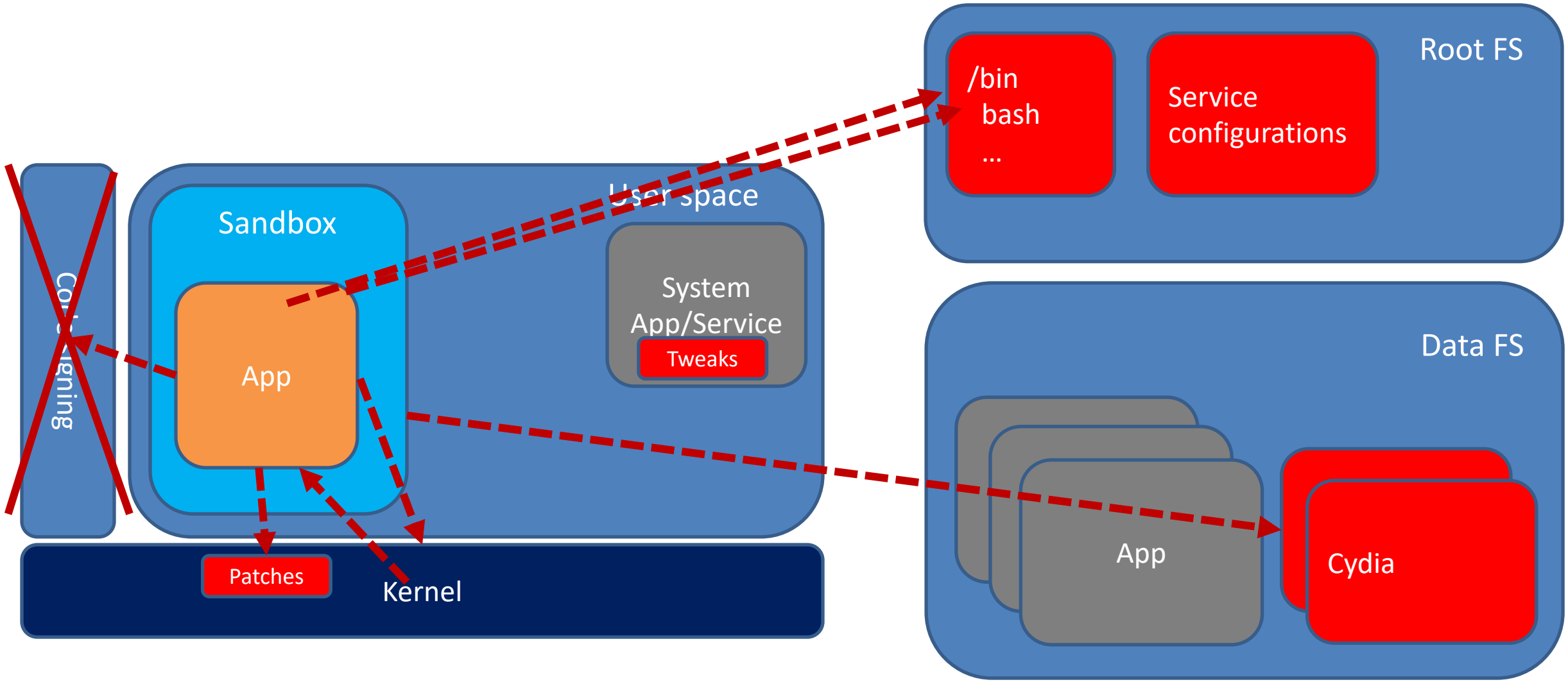
Anatomy of a JB: Remount root FS as R/W



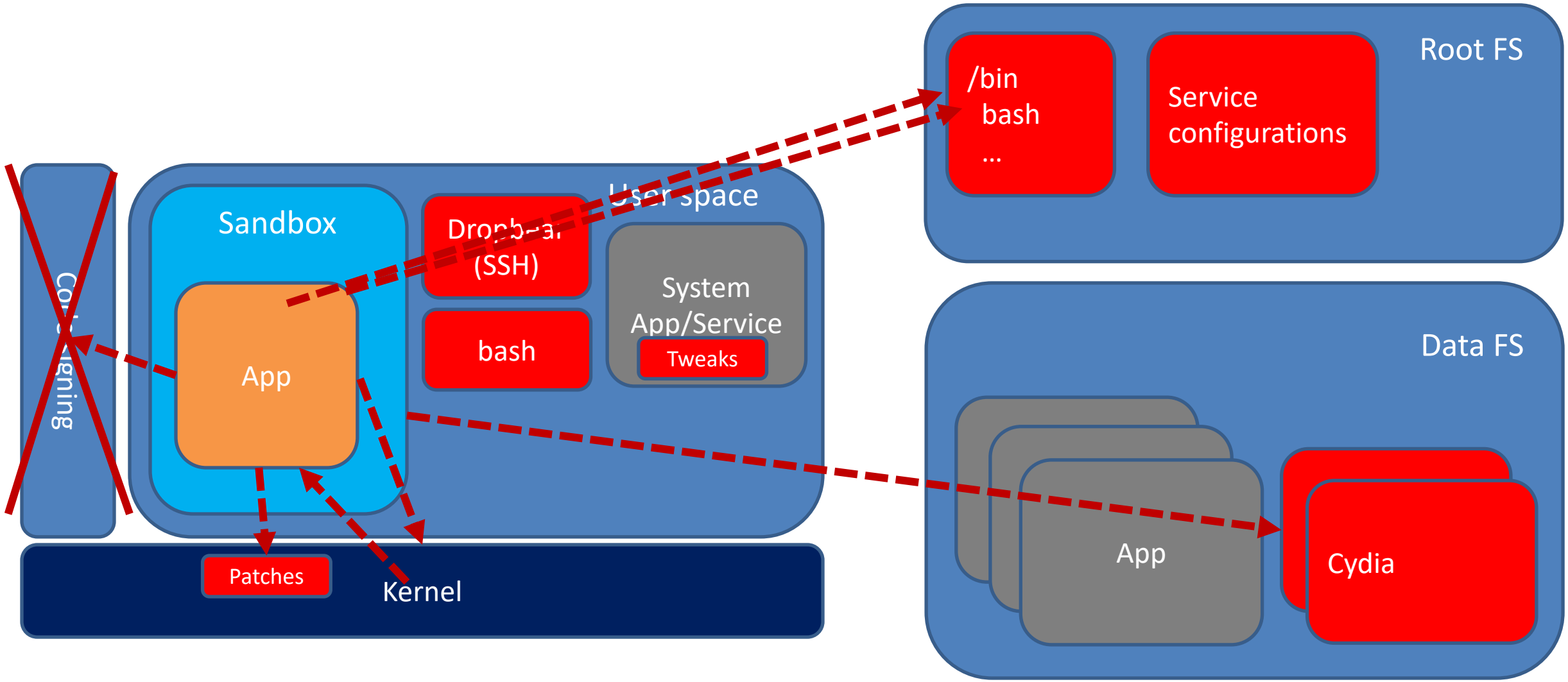
Anatomy of a JB: Extract binaries & Cydia



Anatomy of a JB: Install services & patch binaries



Anatomy of a JB: Install and run SSH/bash



Jailbreaking installs significant amounts of untrusted code on the suspect device

- /Applications/Cydia.app
- /bin and /usr/bin
- /var/stash & /var/lib/cydia - Cydia artefacts
- /var/mobile/Library/Preferences/com.saurik.Cydia.plist.
- /var/MobileDevice/ProvisioningProfiles : provisioning profiles
- /usr/libexec/cydia/*

Other traces include provisioning profiles

bradleys-iPad:/private/var/tmp/bootstrap/bin root# ls -l /var/MobileDevice/ProvisioningProfiles

total 32

-rw-r--r-- 1 mobile mobile 7614 Jan 24 2018 08806c56-9074-4931-86a4-cc162dceb903

-rw-r--r-- 1 mobile mobile 7593 Jan 29 2018 3bcb7785-f9db-4065-94c9-b22350545df3

-rw-r--r-- 1 mobile mobile 7473 Jan 25 2018 71a534c4-d32c-44fc-92c3-d1163a4ca702

-rw-r--r-- 1 mobile mobile 7774 Nov 11 19:03 7b1d1b07-4e32-4a8f-a4f3-0dc4fc273f14

What are the risks of employing regular jailbreaks?

- Uncertain provenance of jailbreak and accompanying 3rd party binaries
- Jailbreak collides with prior jailbreak rendering phone inaccessible
- Jailbreak overwrites traces of prior jailbreak
- Jailbreak leaks information out over network connection
- Jailbreak triggers FindMyiPhone
- Arguments re forensic soundness
- Widespread timestamp overwriting
  - Stashing (OS file relocation) [1]
  - More of an issue with Pangu era jailbreaks
- Partition resizing ?

- [1] https://www.theiphonewiki.com/wiki//private/var/stash

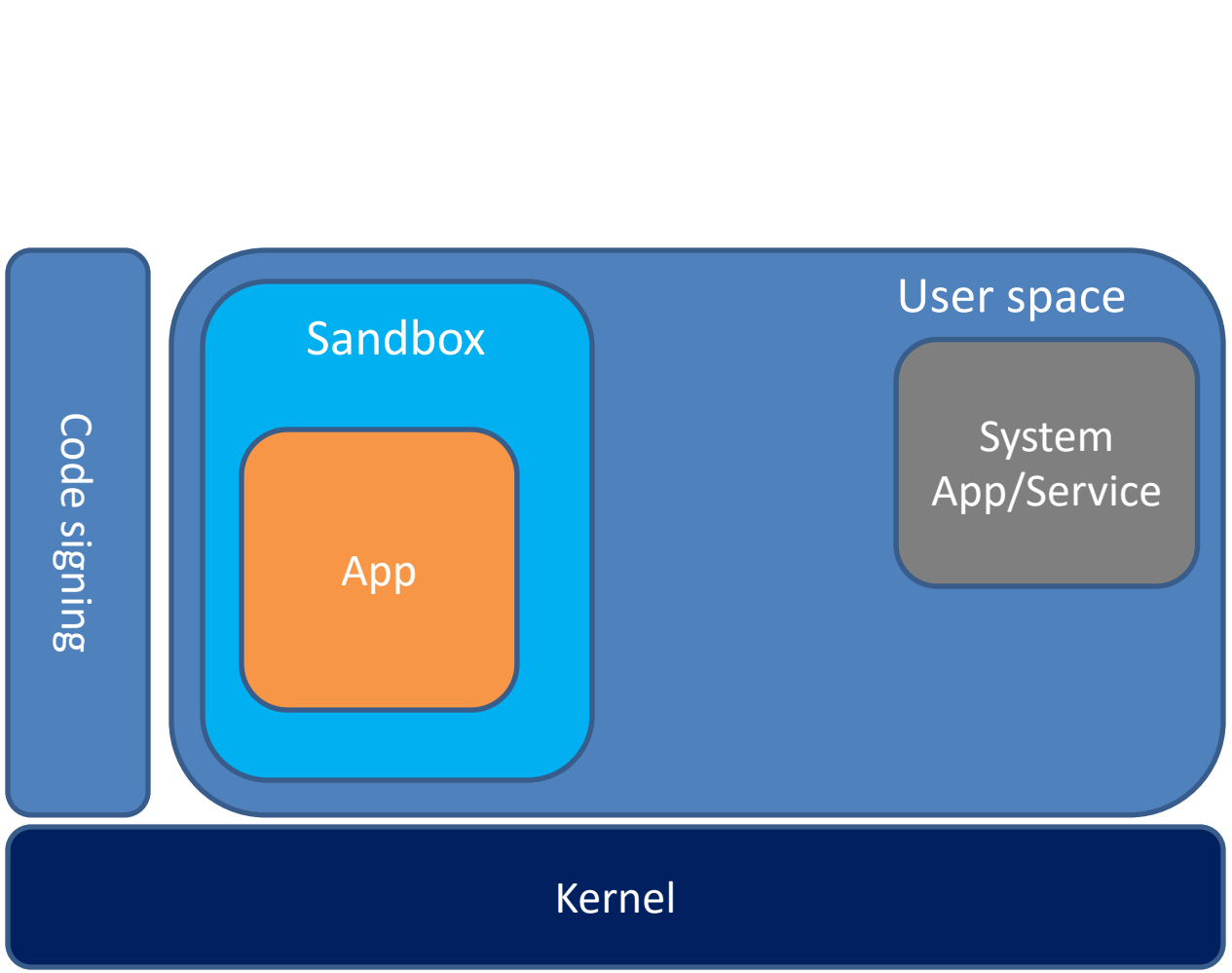
A proposal – forensic jailbreaking

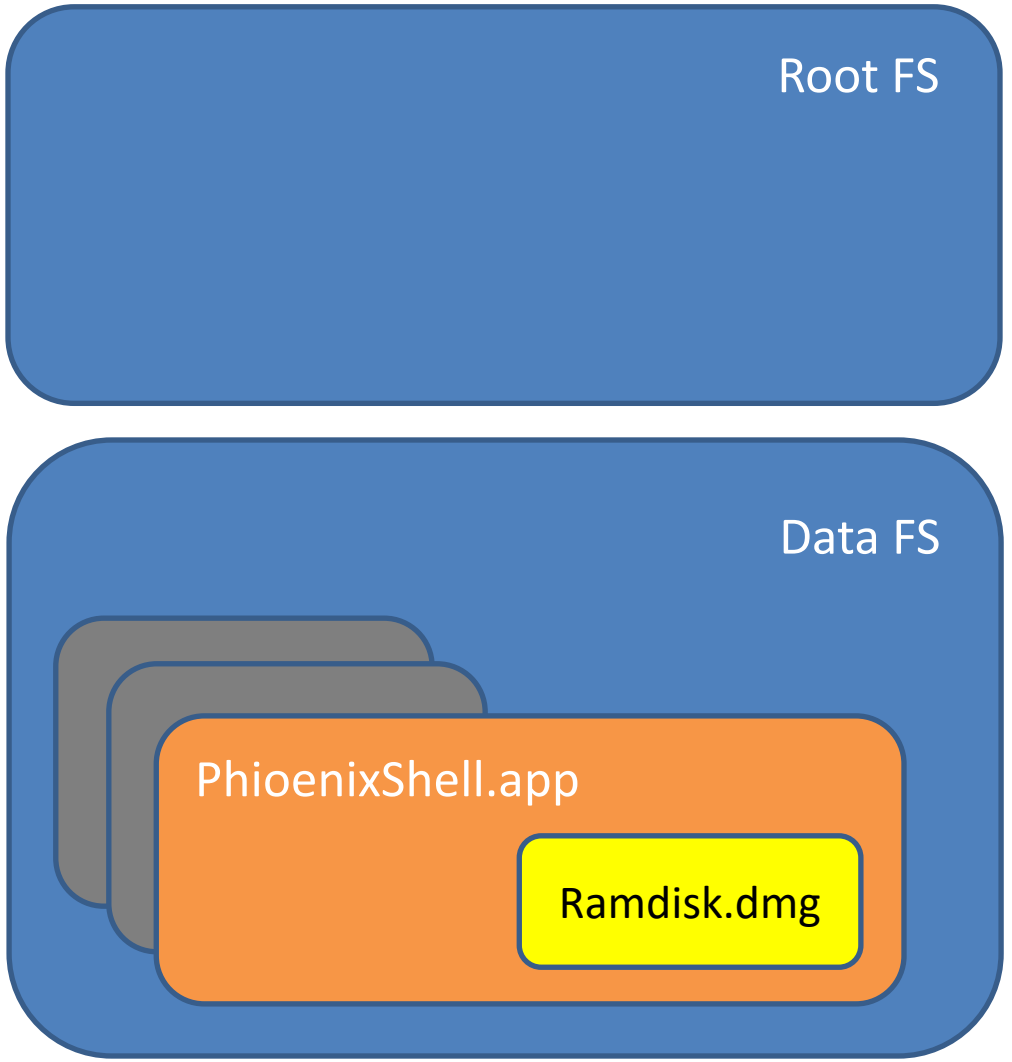
Forensic jailbreak prototype 1

- Overriding goals:
  - Minimise/quantify changes to filesystem
    - Don’t remount root as R/W
  - Don’t overwrite existing jailbreak traces
  - Don’t collide with existing jailbreaks (eg TCP listening port)

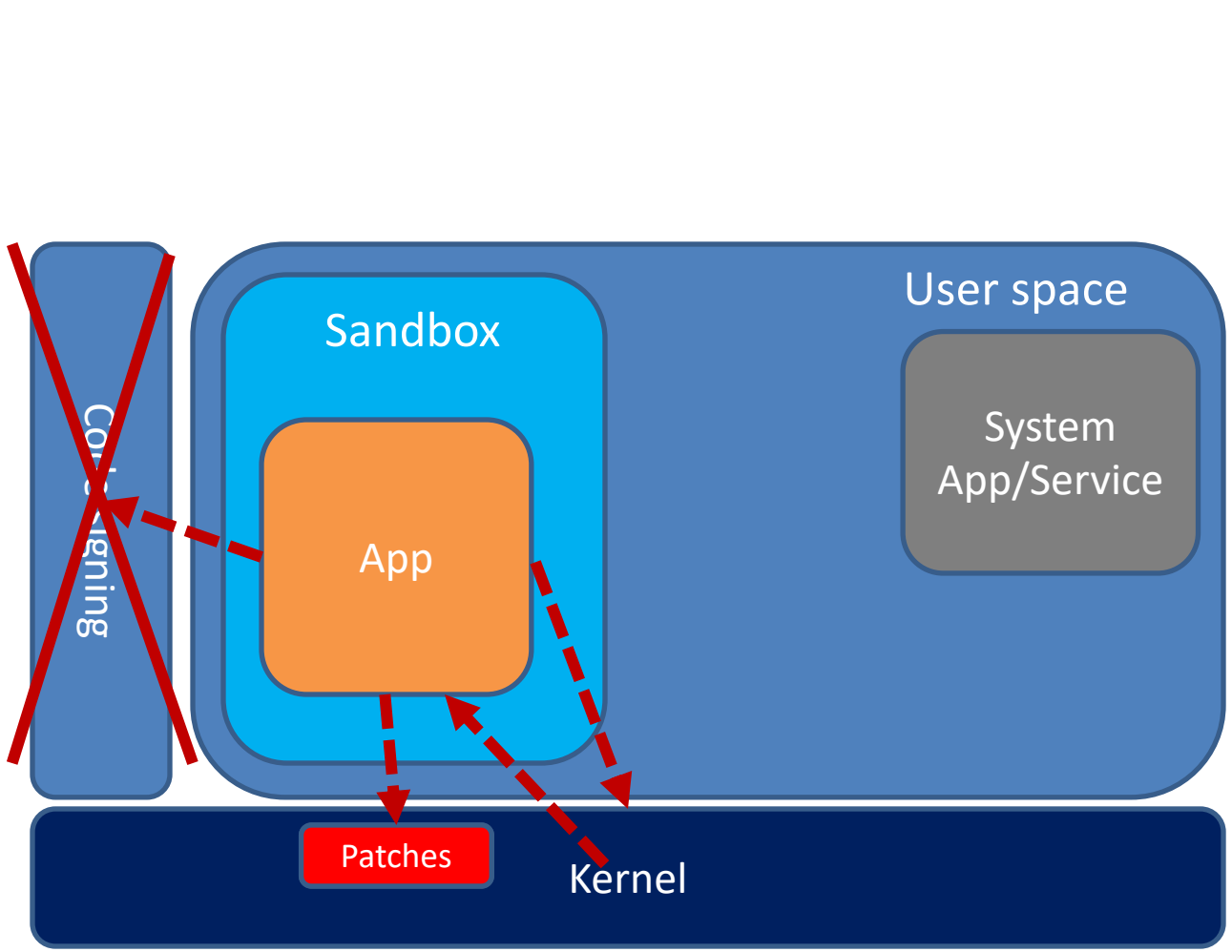
- Theory
  - Load minimal SSH server and rely on SFTP for file enumeration/copy

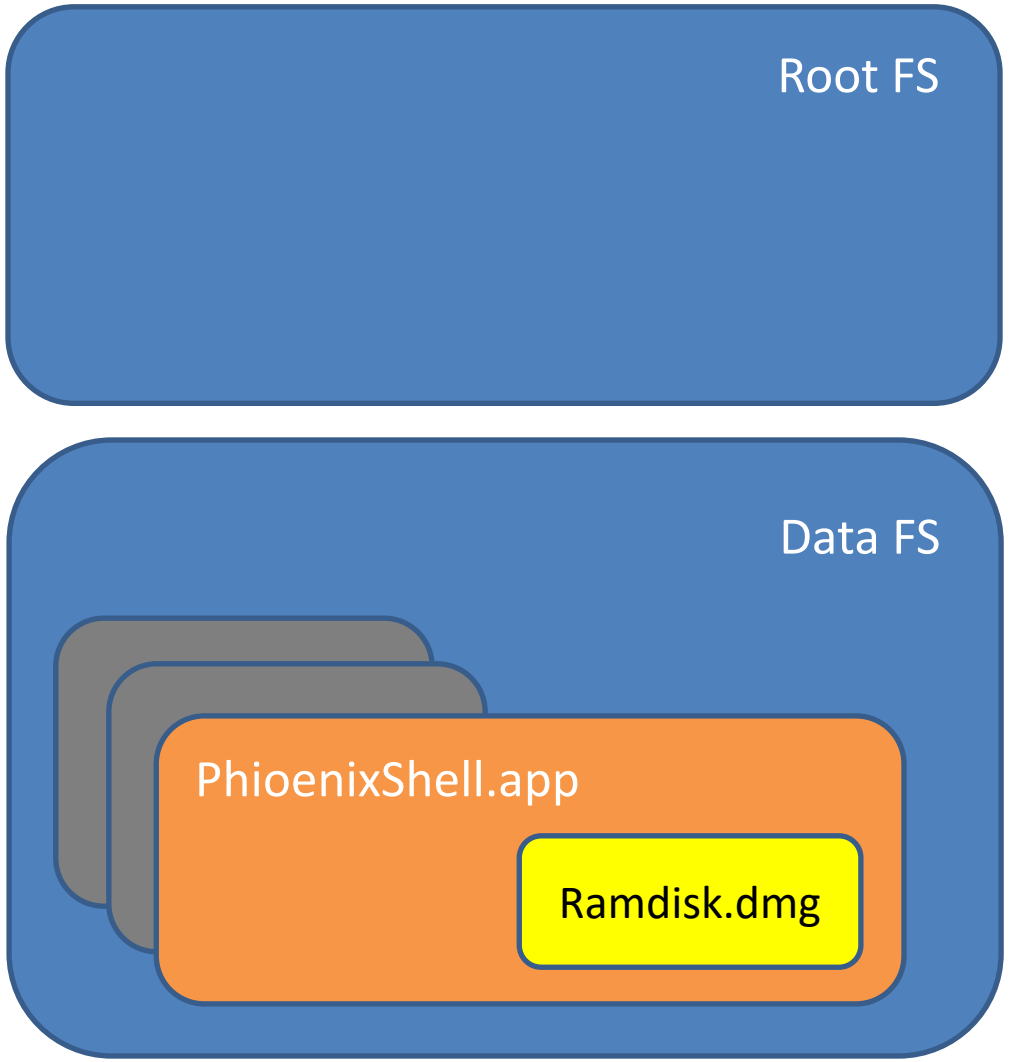
Forensic JB Prototype 1



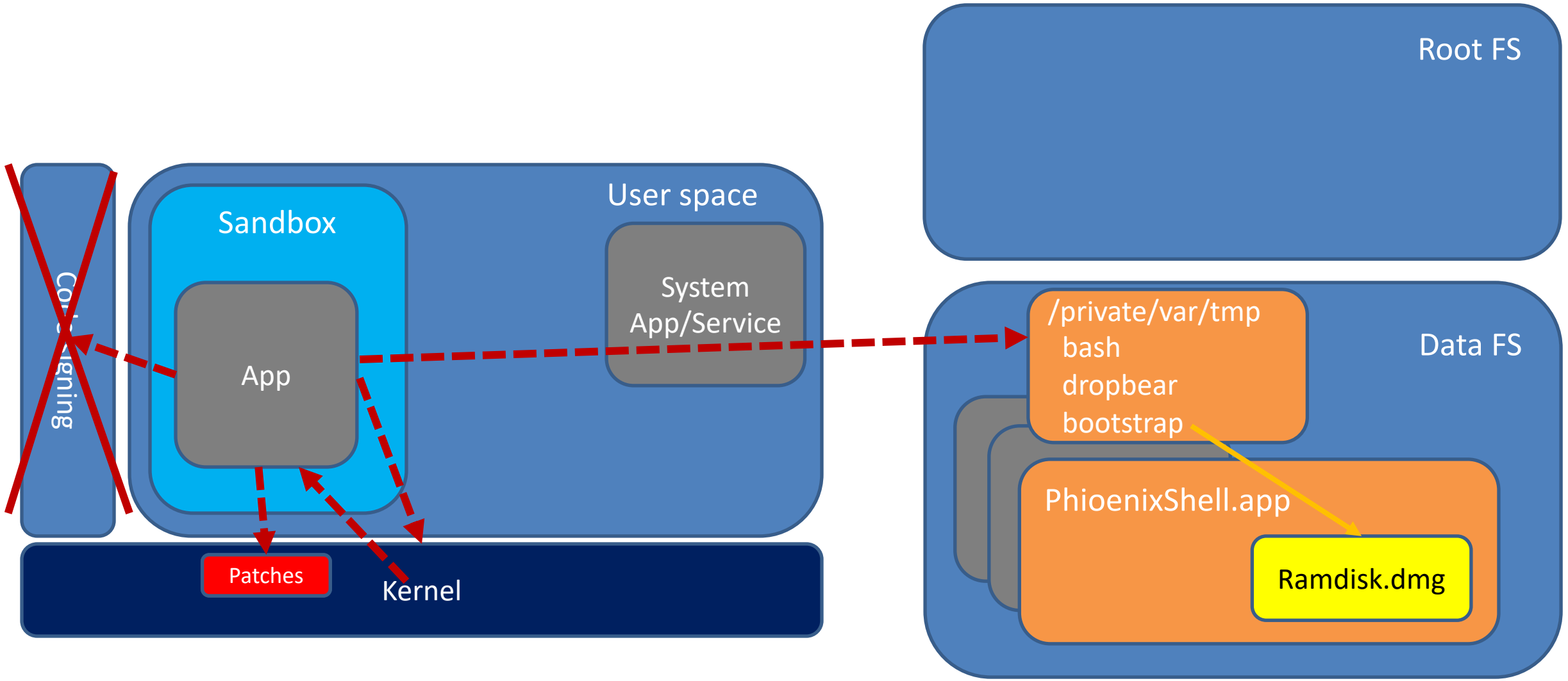


Prototype 1: Regular first stage

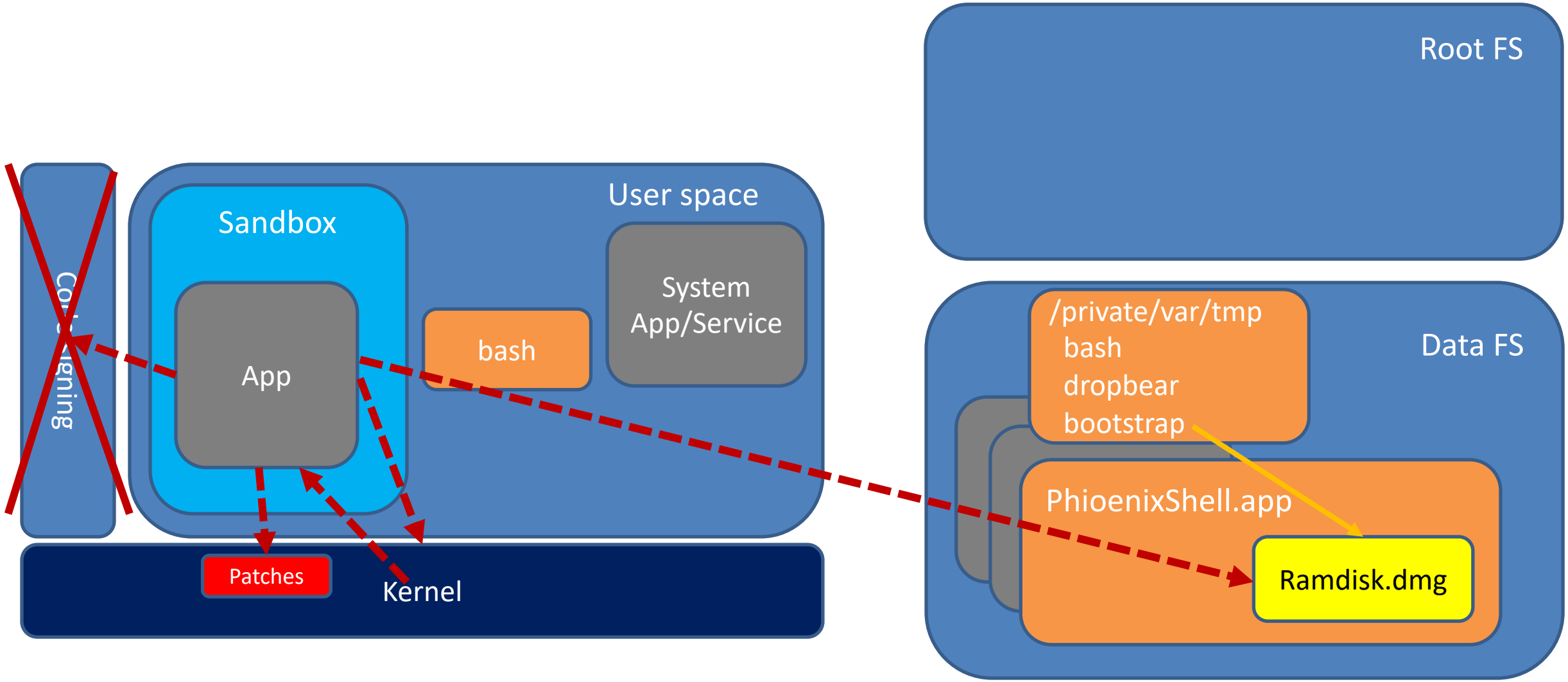




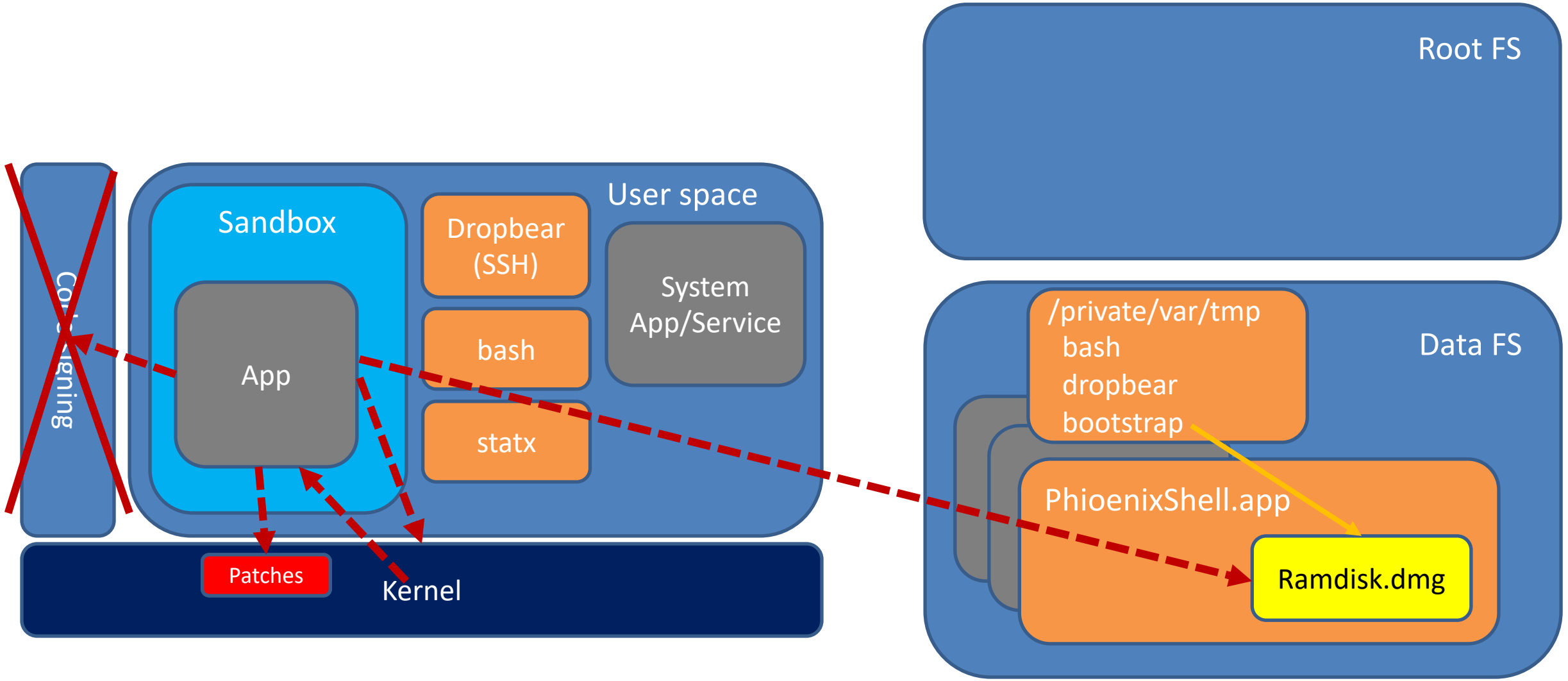
Prototype 1: Extract exe's and mount ramdisk



Prototype 1: Bind bash to TCP port



Prototype 1: Run SSH server, and acquire



Forensic jailbreak prototype 1
persistent changes made

- Load app on device:
  - new name - PhoenixShell.app
- Extract to /private/var/tmp/
  - bash
  - dropbear
- Create folder under /private/var/tmp/
  - bootstrap (PhoenixShell.app/bootstrap.dmg mounted here)

Forensic jailbreak prototype 1:
Client side usage

- In one shell

```
neon:~ bradley$ iproxy 4444 44
waiting for connection
```

- In another

```
neon:pyaff4 bradley$ nc localhost 4444
bradleys-iPad:/ root#
```

- Manually
  - Use ssh, tar, stat for examination

Forensic jailbreak prototype 1

Automated client side acquisition

- Establish python/paramiko SSH connection
- Upload stat to tmp folder on device using unique name
- Enumerate filesystem metadata and store in AFF4 image
- Supplement filesystem metadata with file creation time metadata from stat
- Copy file content using SCP into AFF4 image

Forensic jailbreak prototype 1

AFF4 Logical Image Contents

```
neon:phoenixShell bradley$ unzip -l /tmp/iPad.aff4
Archive:  /tmp/iPad.aff4
aff4://2b1e5aae-b7cf-42f4-bdcb-c1c8aa4e94ab
  Length      Date    Time    Name
---------  ---------- -----   ----
   154048  00-00-1980 00:00   /usr/bin/brctl
    52240  00-00-1980 00:00   /usr/bin/arch
    87856  00-00-1980 00:00   /usr/bin/captoinfo
    49856  00-00-1980 00:00   /usr/bin/cfversion
     4374  00-00-1980 00:00   information.turtle
     6558  00-00-1980 00:00   /usr/bin/apt-key
     3667  00-00-1980 00:00   /usr/bin/c_rehash
     6822  00-00-1980 00:00   /usr/bin/bashbug
       28  00-00-1980 00:00   version.txt
       43  00-00-1980 00:00   container.description
---------                     -------
   365492                     10 files
```

Forensic jailbreak technique – try #1:

AFF4 Logical Metadata

```
<aff4://685faddc-be15-429e-b240-6bd002e1196b//.fseventsd/00000000002a9441> a aff4:FileImage,
        aff4:Image ;
    aff4:birthTime "2018-11-23T16:33:48+10:00"^^xsd:datetime ;
    aff4:hash "ff928ebb6fc2efcf6f7d02619c3d832a"^^aff4:MD5,
        "61e09a12ff94516d334ac311e4c08144f37604bc"^^aff4:SHA1 ;
    aff4:lastAccessed "2018-11-23T16:33:48+10:00"^^xsd:datetime ;
    aff4:lastWritten "2018-11-23T16:33:48+10:00"^^xsd:datetime ;
    aff4:originalFileName "/.fseventsd/00000000002a9441"^^xsd:string ;
    aff4:recordChanged "2018-11-23T16:33:48+10:00"^^xsd:datetime ;
    aff4:size 23047 .
```

Forensic jailbreak prototype 1:
Limitations

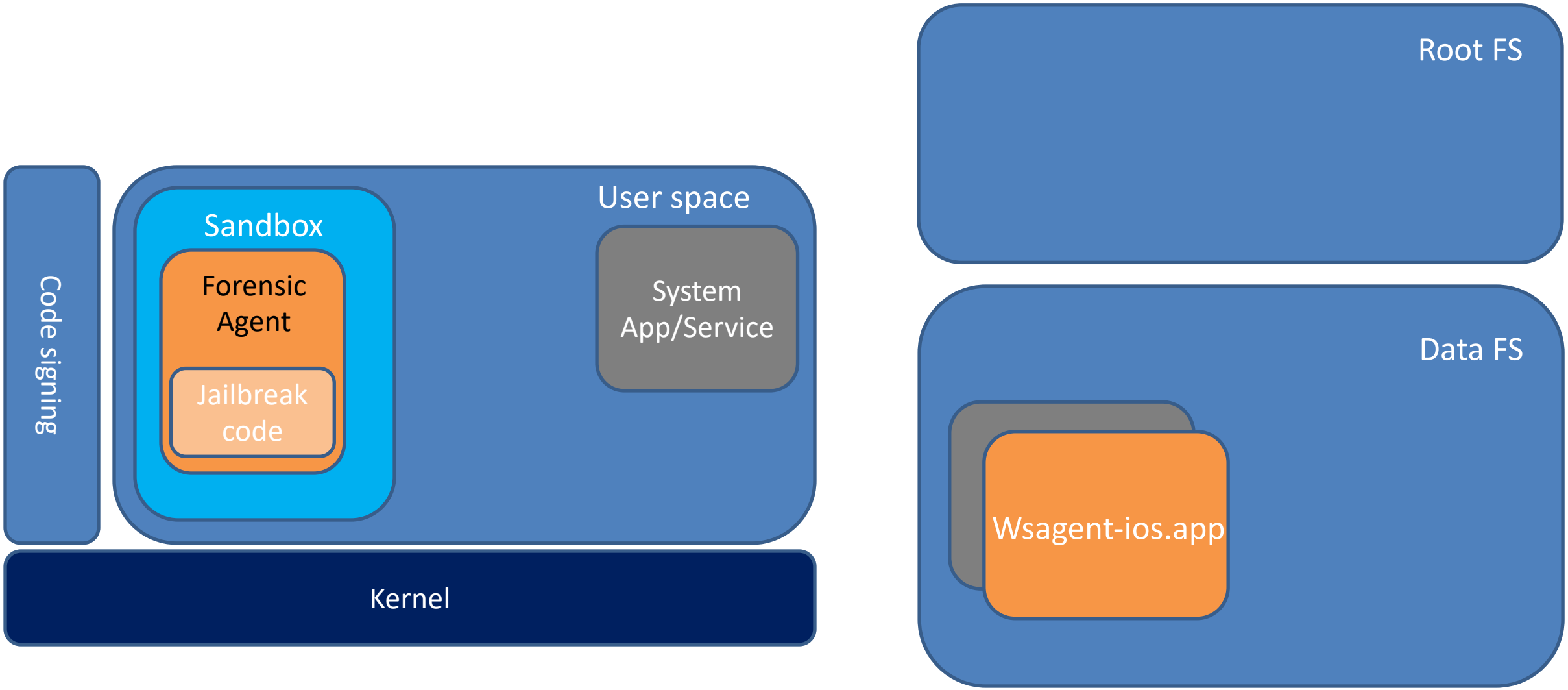
- Needs complex jailbreak to run SSH server, bash & stat
- Uncertain operation in presence of still-running jailbreak
- Medium impact on changes to filesystem
- Potential trigger of *Find my iPhone*

An improved proposal – forensic jailbreaking

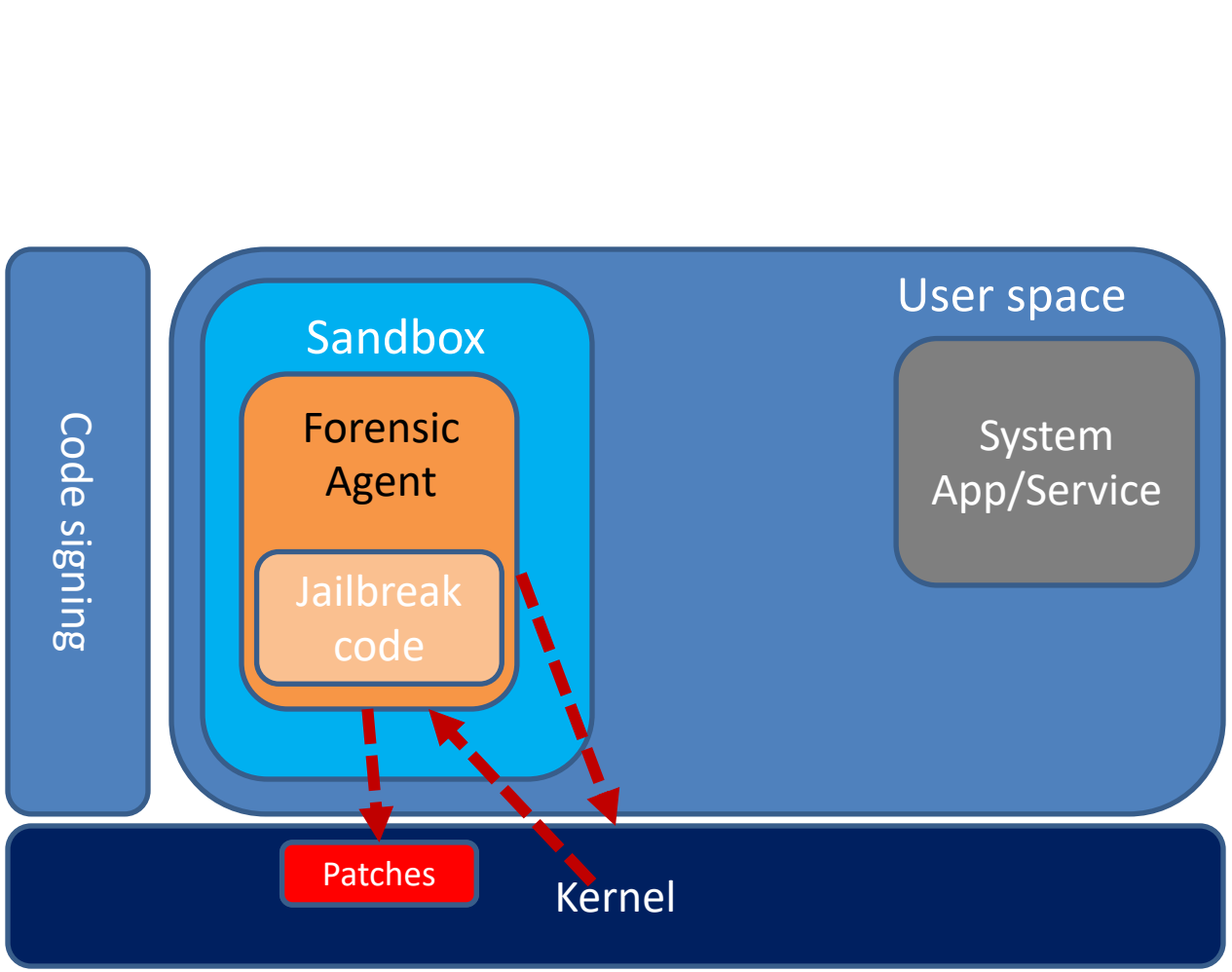
Forensic jailbreak prototype 2:

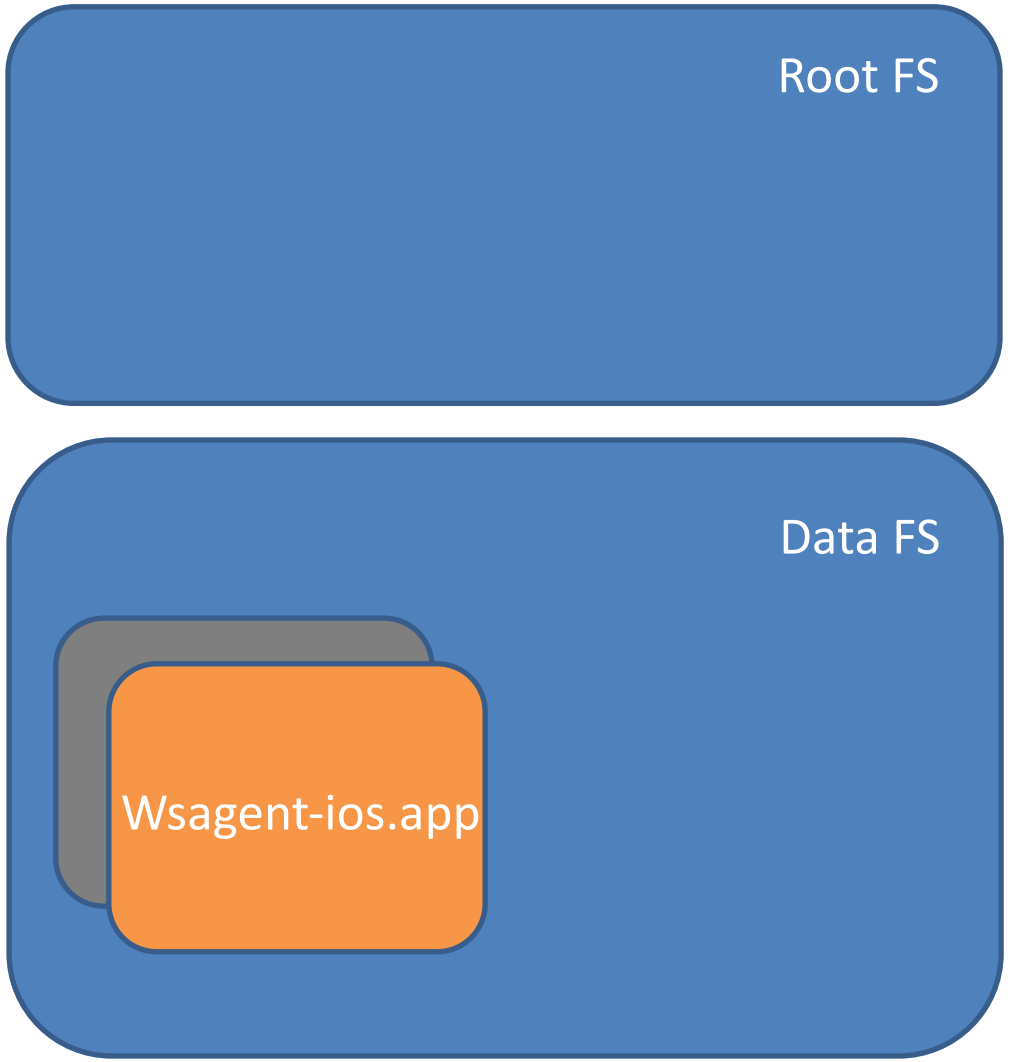
- Only use a single process supporting
  - Jailbreak
  - Access (Files, Disk, RAM)
- Benefits
  - Less complex jailbreak needed
  - No third party binaries needed
  - Minimal impact on suspect filesystem

Forensic Jailbreak Prototype 2



Forensic Jailbreak Prototype 2





Agent in action

Future work

- Keychain decryption
- Physical storage (APFS + encryption)
- RAM

FIN

- Thanks to:
  - Jonathan Levin
  - @siguza (Forensic jailbreak 1 collaborator)
  - @undecimus for unc0ver jailbreak
  - @i41nbeer, @bazad, @xerub, @iBSparkes, @stek29, @theninjaprawn, @FCE365 for the underlying jailbreak techniqes

Contact

Dr Bradley Schatz
https://evimetry.com/
bradley@evimetry.com
@blschatz