How iOS Security Really Works
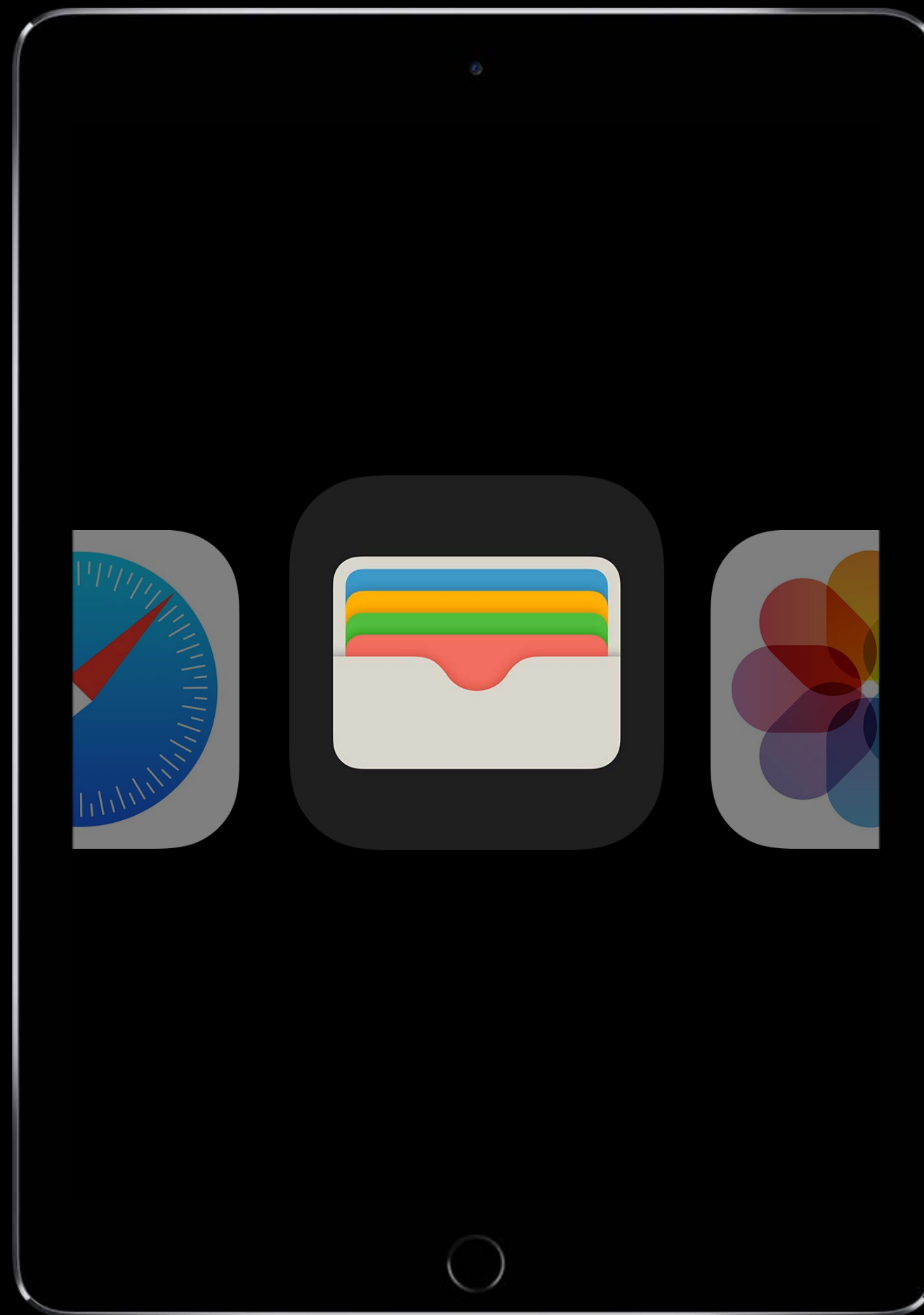
Session 705

Ivan Krstić Head of Security Engineering & Architecture

Mobile Devices Today

Unprecedented record of our lives
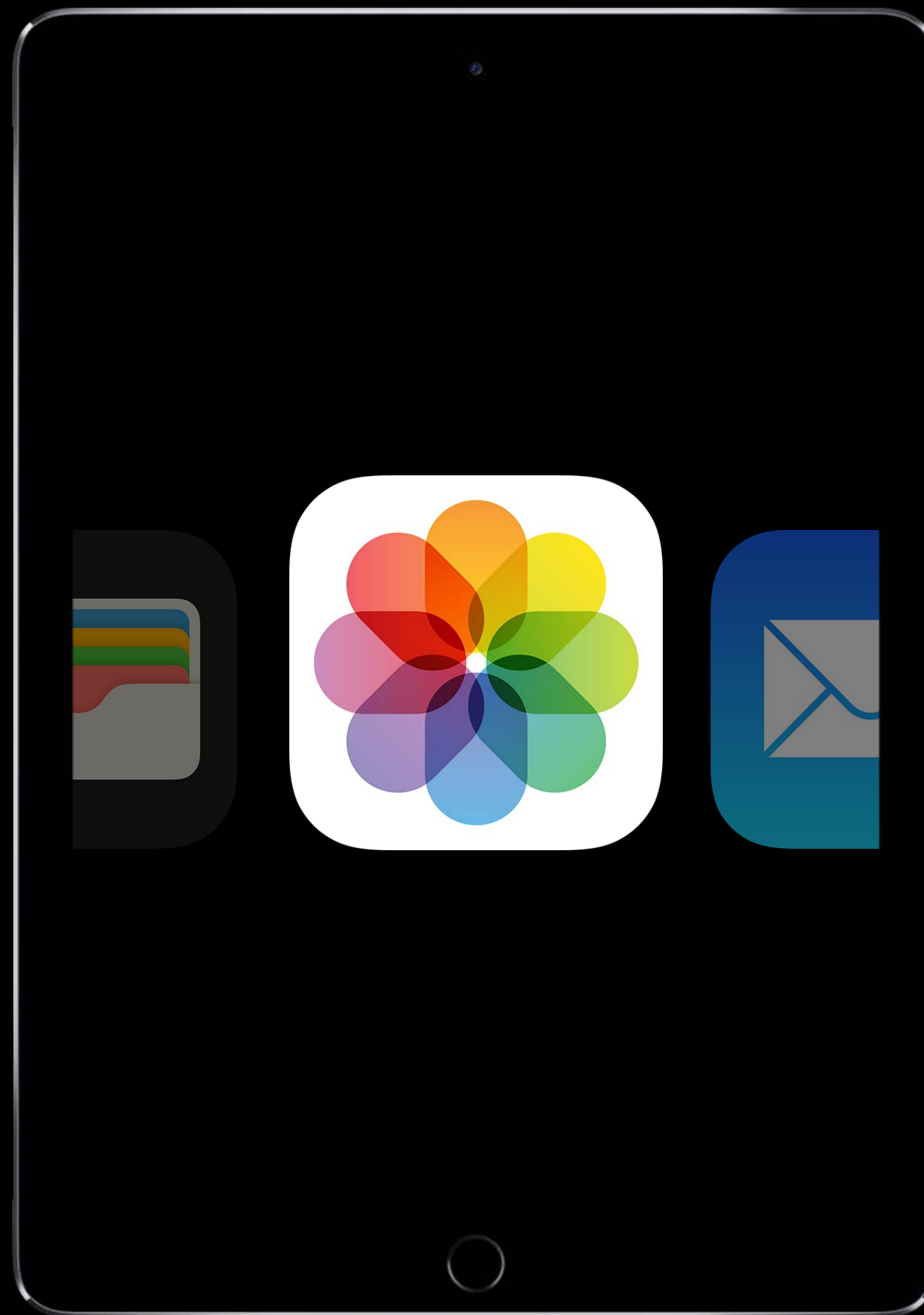

Gb of Data

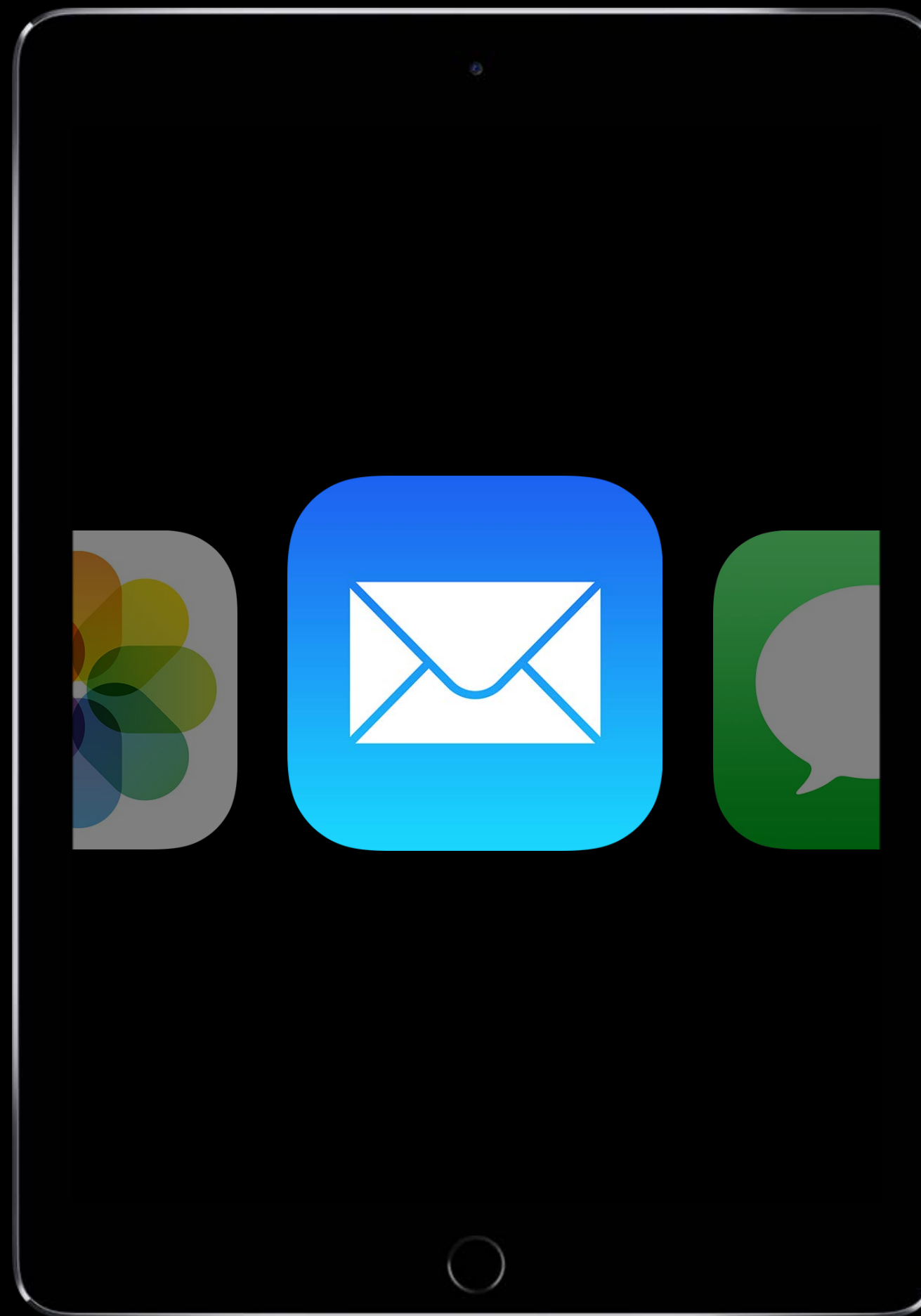# Mobile Devices Today

Unprecedented record of our lives

Gb of Data

Mobile Devices Today

Unprecedented record of our lives

Gb of Data

Mobile Devices Today
Unprecedented record of our lives

Gb of Data

Who Might Be an Attacker

Who Might Be an Attacker

Criminals

Who Might Be an Attacker

Criminals

Business competitors

Who Might Be an Attacker

Criminals

Business competitors

Service providers

Who Might Be an Attacker

Criminals

Business competitors

Service providers

Nation states

Who Might Be an Attacker

Criminals

Business competitors

Service providers

Nation states

Romantic partners, family, friends

Who Might Be an Attacker
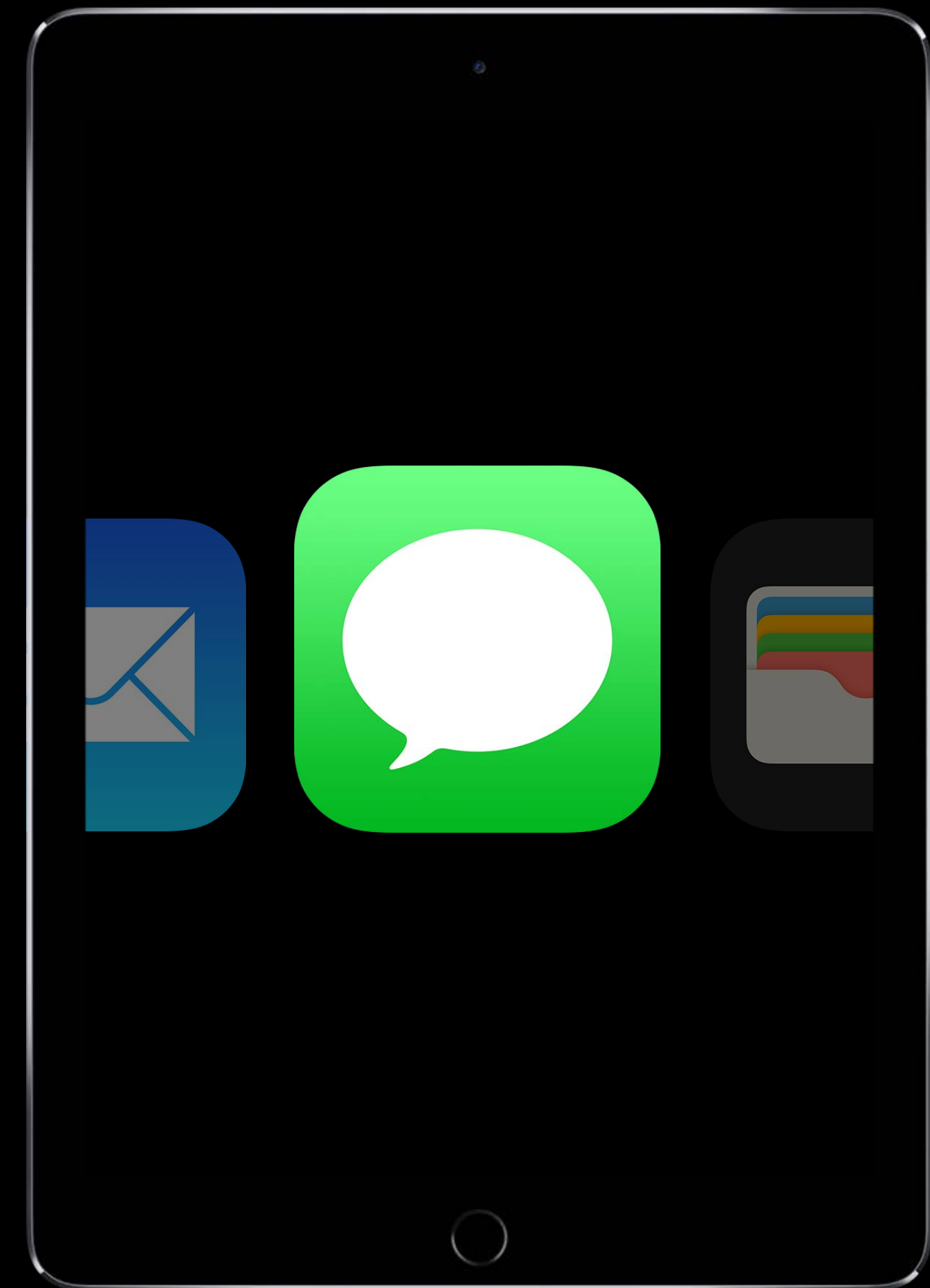
Criminals

Business competitors

Service providers

Nation states

Romantic partners, family, friends

Cats

What Do Attackers Want?

What Do Attackers Want?

Personal stalking and surveillance

What Do Attackers Want?

Personal stalking and surveillance
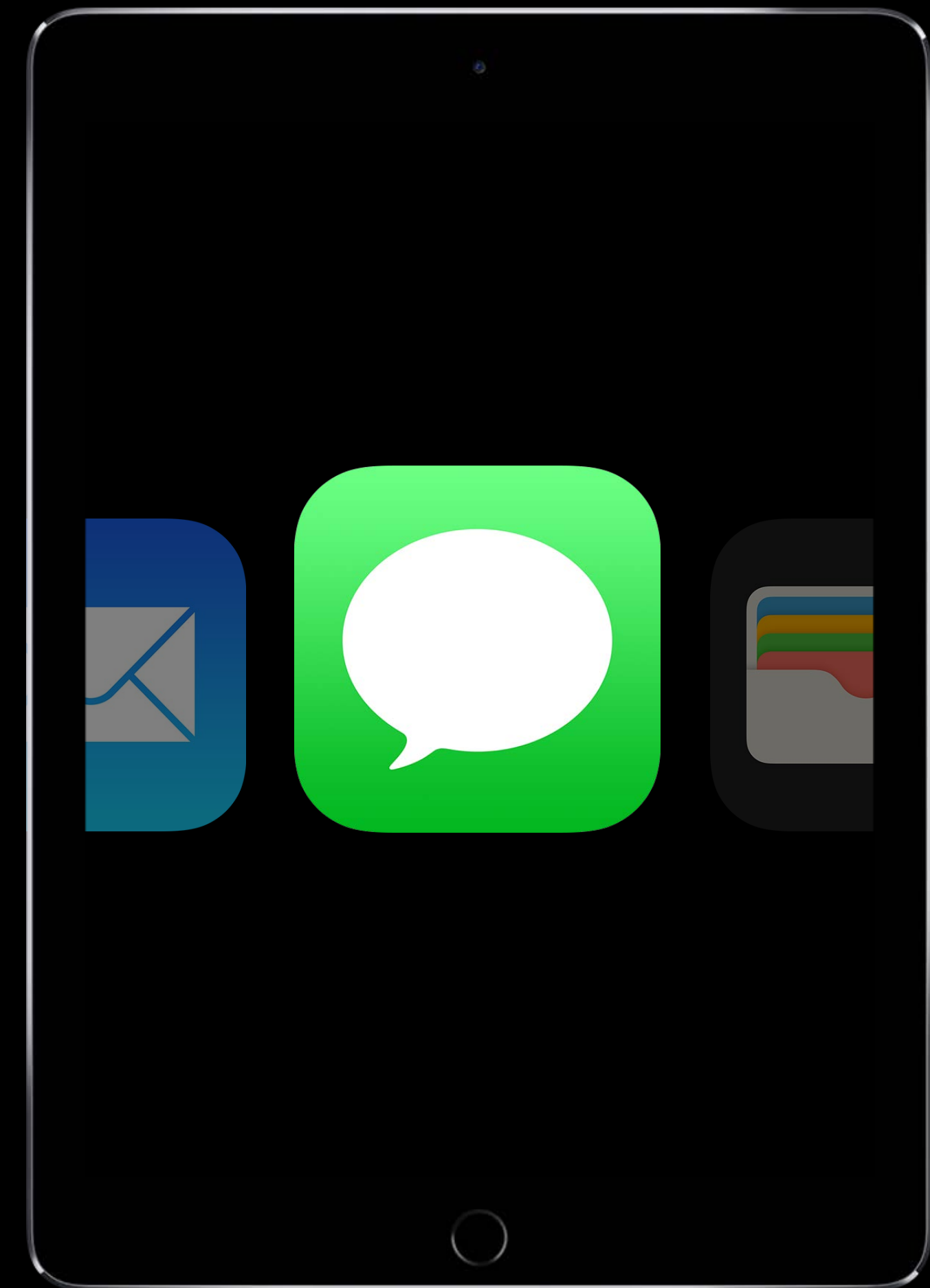
Corporate espionage

What Do Attackers Want?

Personal stalking and surveillance

Corporate espionage

Direct financial benefit

How Do We Know This?

We see it on other platforms

But Not on iOS

No malware has affected iOS
devices at scale

iOS Security

iOS Security

Decade-long effort to protect customers from security problems

iOS Security

Decade-long effort to protect customers from security problems

Incredible scale

iOS Security

Decade-long effort to protect customers from security problems

Incredible scale

Every single iOS security feature is designed to thwart a real security threat

iOS Security Pillars

iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

Traditional Security

physical security

secure configuration

installing latest patches

password policy

vetted apps

mandated policies

Traditional Security	iOS Security
physical security	security built from silicon up
secure configuration	secure default settings
installing latest patches	easy updates
password policy	Touch ID
vetted apps	App Store
mandated policies	ease of use

1. Secure Boot

2. Data Protection

3. Sandboxing

4. Code Signing

5. Touch ID

① Secure Boot

② Data Protection
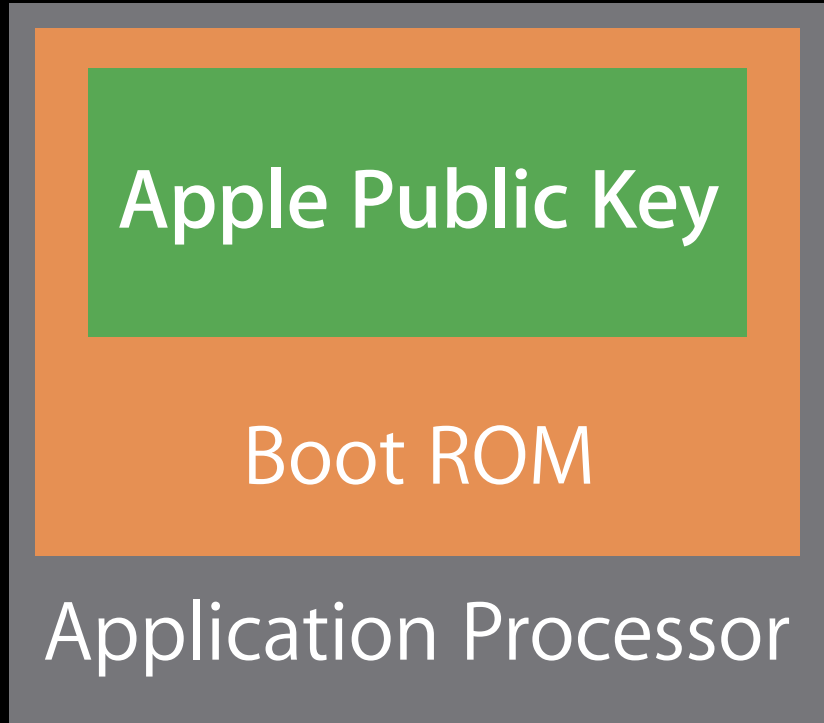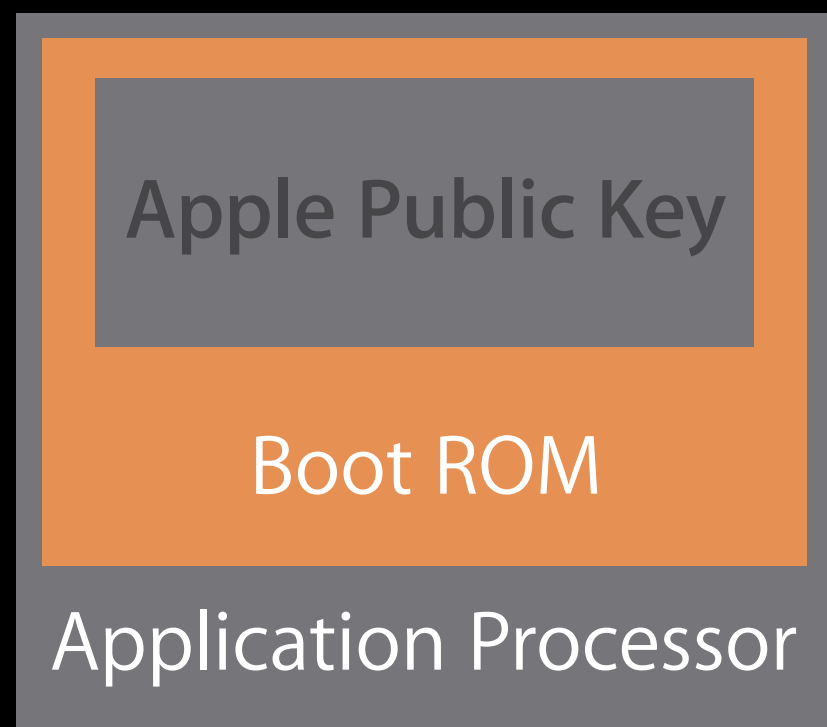
③ Sandboxing

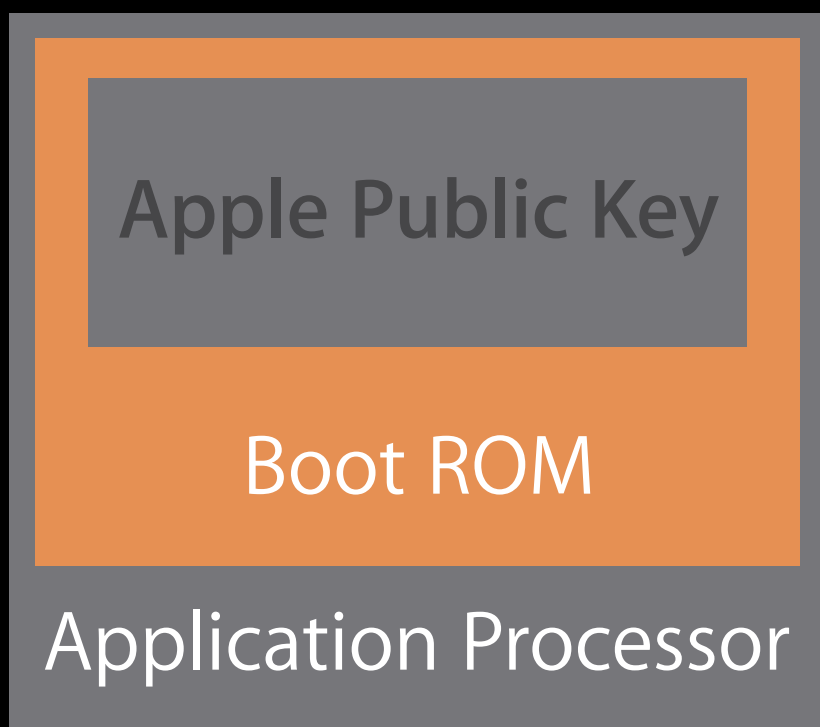④ Code Signing

⑤ Touch ID
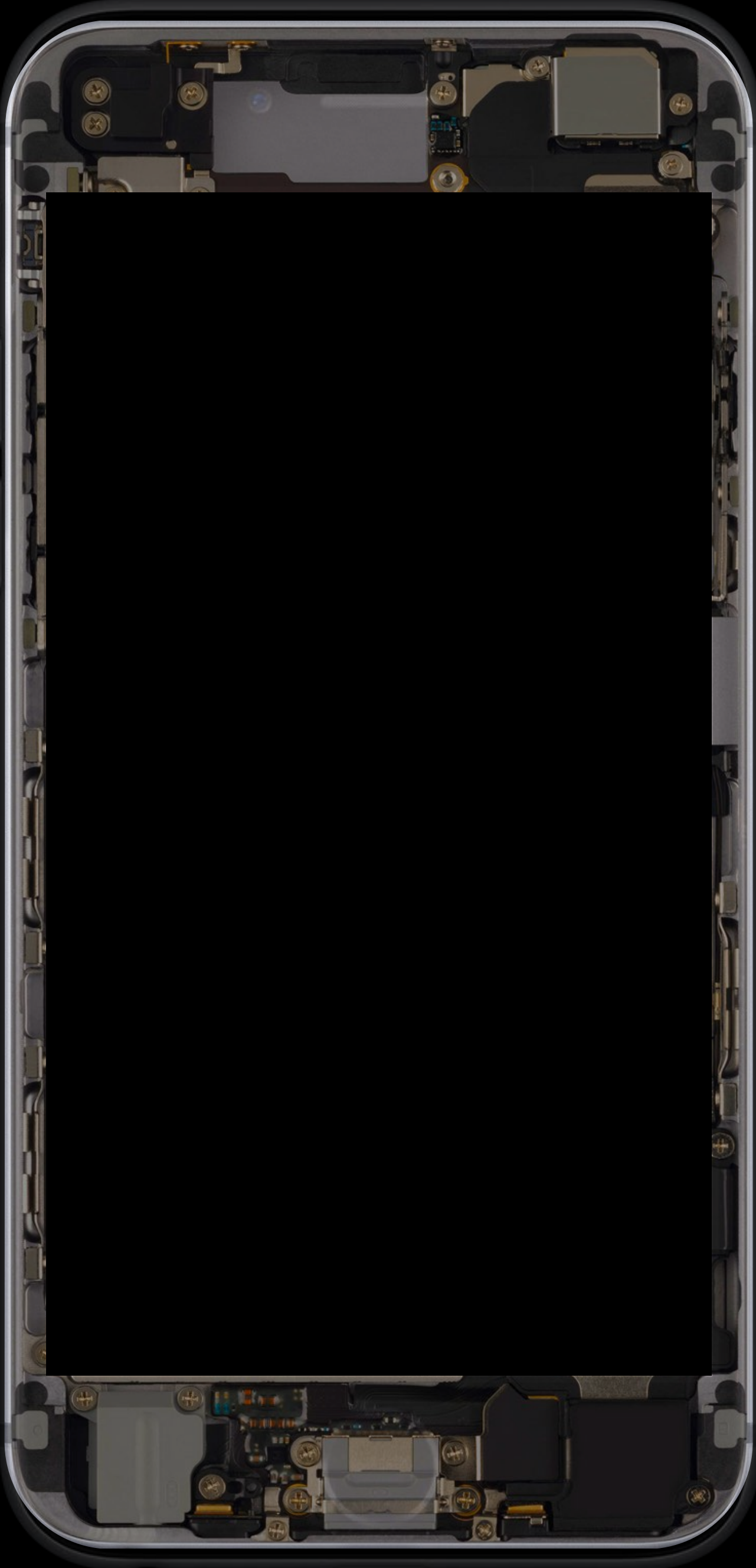
Trust built from silicon up

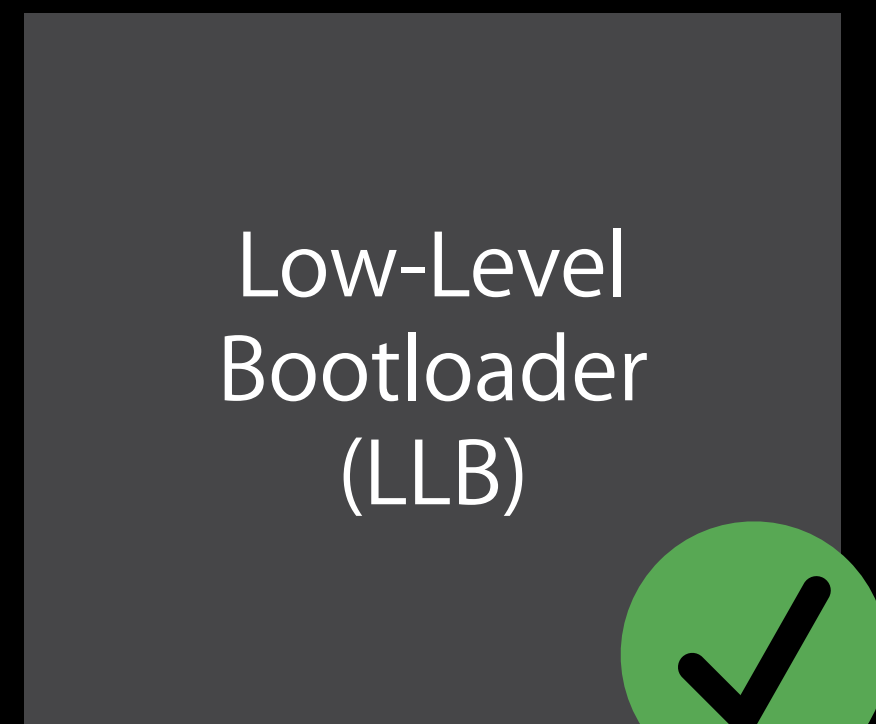Application Processor
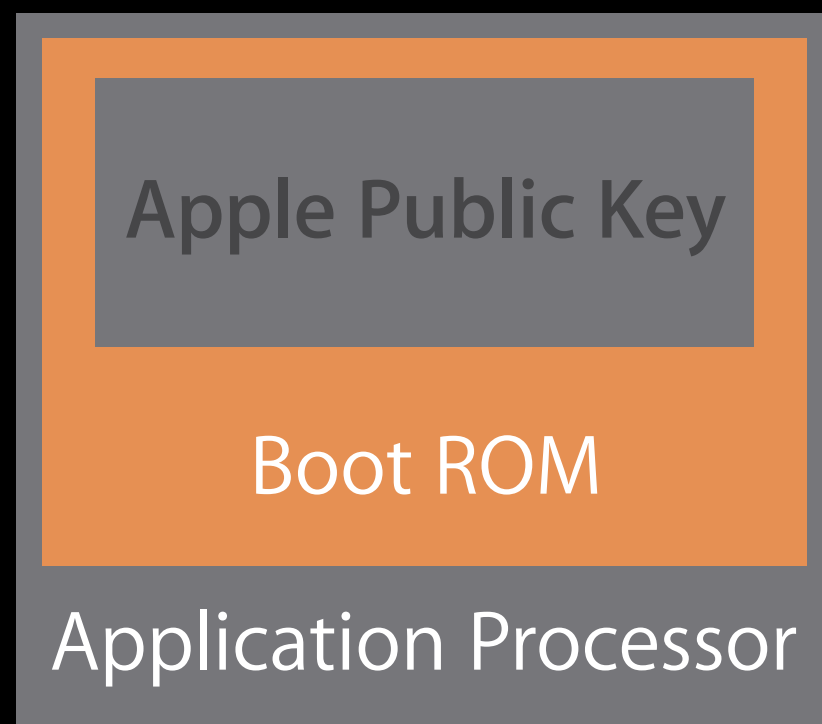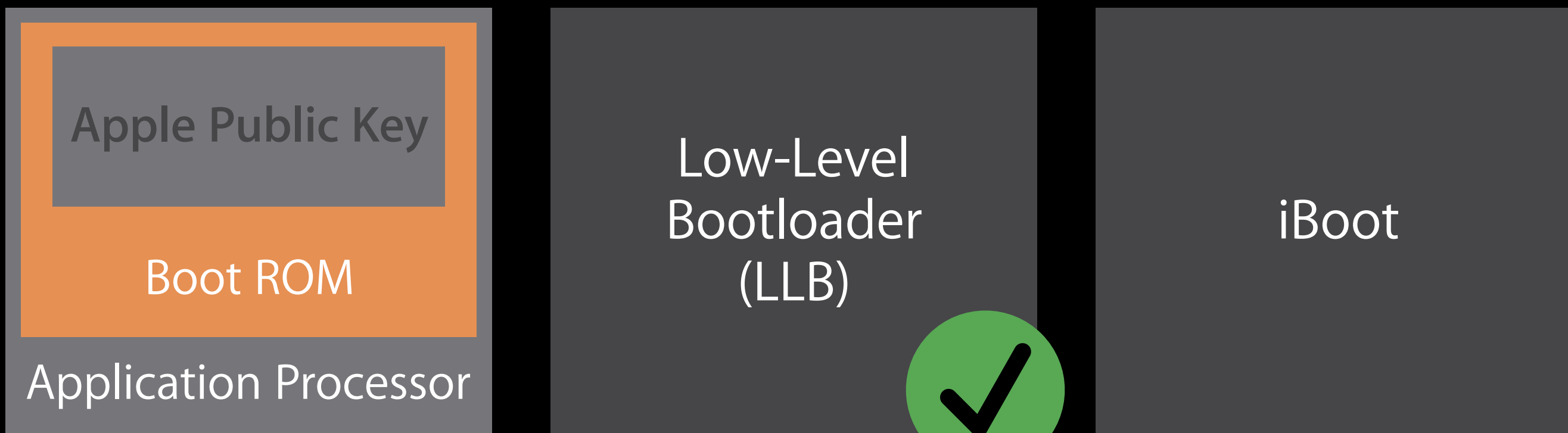
Boot ROM
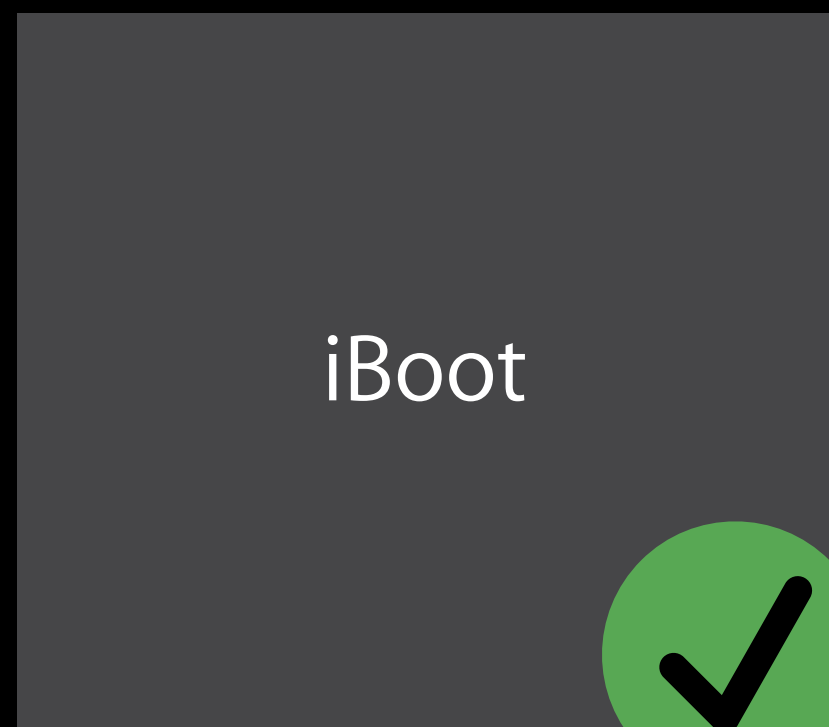
Application Processor

Apple Public Key

Boot ROM

Application Processor

Apple Public Key

Apple Public Key

Apple Public Key

Application Processor

Apple Public Key

Boot ROM

Low-Level Bootloader (LLB) ✔

iBoot ✔

Kernel ✔

iOS ✔
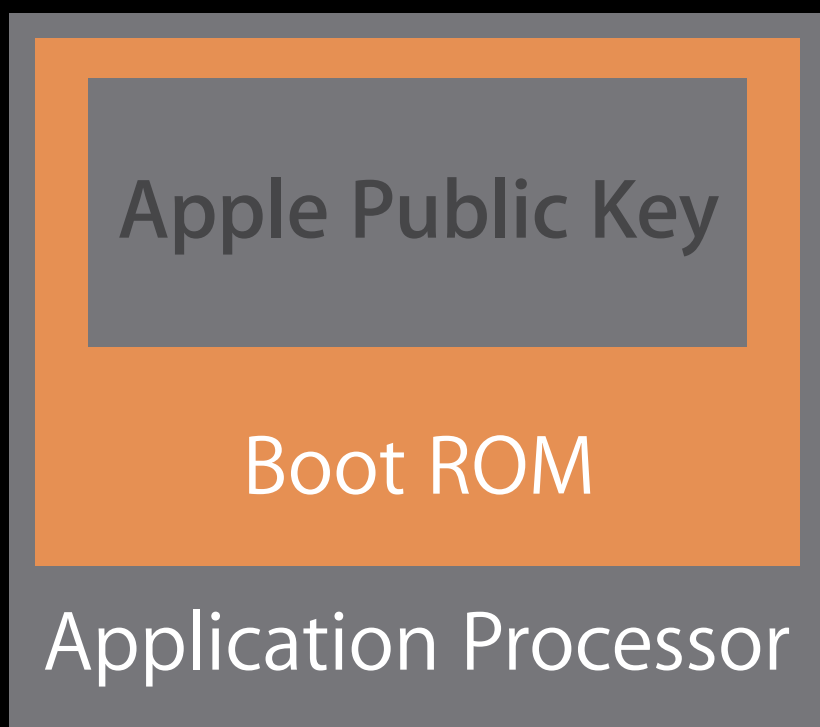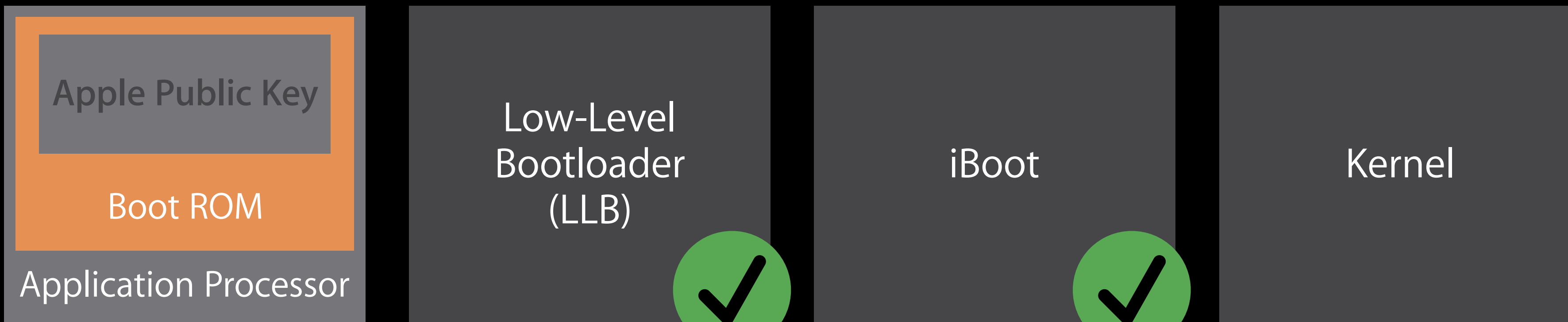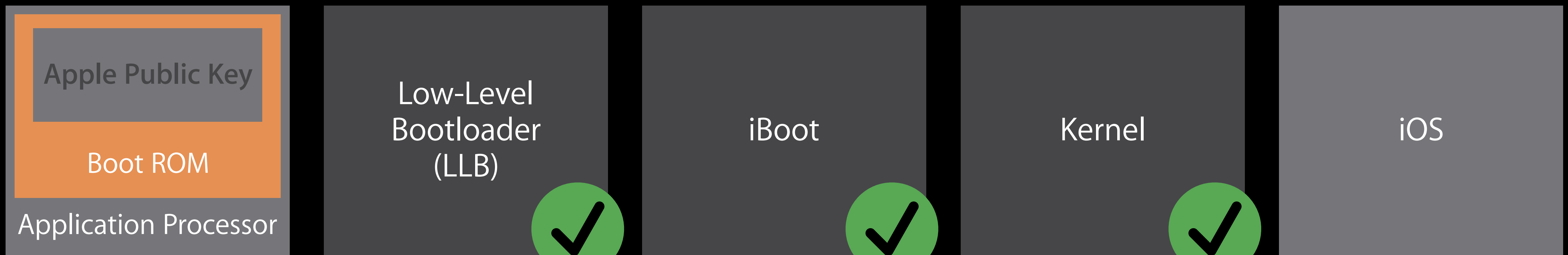
Apple Public Key
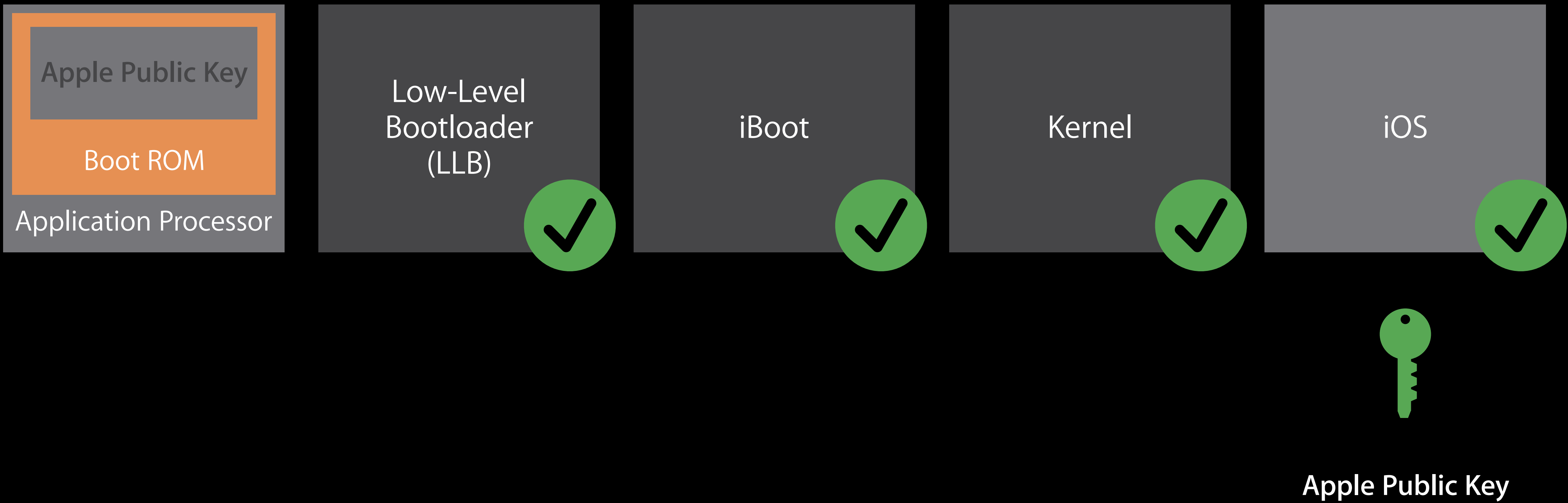
Trusting Secure Boot

Trusting Secure Boot

Keys are securely provisioned and managed by Apple

Trusting Secure Boot

Keys are securely provisioned and managed by Apple

Software updates are authorized individually for each device

1. Secure Boot
2. Data Protection
3. Sandboxing
4. Code Signing
5. Touch ID

1. Secure Boot
2. **Data Protection**
3. Sandboxing
4. Code Signing
5. Touch ID

Data Protection

Data Protection

User data is encrypted at rest with keys
derived from the passcode

Data Protection

User data is encrypted at rest with keys derived from the passcode

Entangled with hardware key in SEP

Data Protection

Data Protection

SEP refuses to unlock after more than 10 incorrect passcode attempts

Data Protection

SEP refuses to unlock after more than 10 incorrect passcode attempts

'Erase Data' only controls erasure, not ability to unlock

Standard algorithms

Internal security audits

3rd-party code review

Cryptographic Libraries

The same libraries that secure iOS and OS X are available to third-party developers to help them build advanced security features.

Security Framework

Security Framework provides interfaces for managing certificates, public and private keys, and trust policies. It supports the generation of cryptographically secure pseudorandom numbers. It also supports the storage of certificates and cryptographic keys in the keychain, which is a secure repository for sensitive user data.

📄 iOS Security Framework Reference
📄 OS X Security Framework Reference
📁 Security Framework on Apple Open Source
💬 Apple Developer Forums: Security

Common Crypto

The Common Crypto library provides additional support for operations like symmetric encryption, hash-based message authentication codes, and digests.

📄 Cryptographic Services Guide
📁 Common Crypto on Apple Open Source

corecrypto

Both Security Framework and Common Crypto rely on the corecrypto library to provide implementations of low level cryptographic primitives. This is also the library submitted for validation of compliance with U.S. Federal Information Processing Standards (FIPS) 140-2 Level 1. Although corecrypto does not directly provide programming interfaces for developers and should not be used by iOS or OS X apps, the source code is available to allow for verification of its security characteristics and correct functioning.

⬇ Download corecrypto source

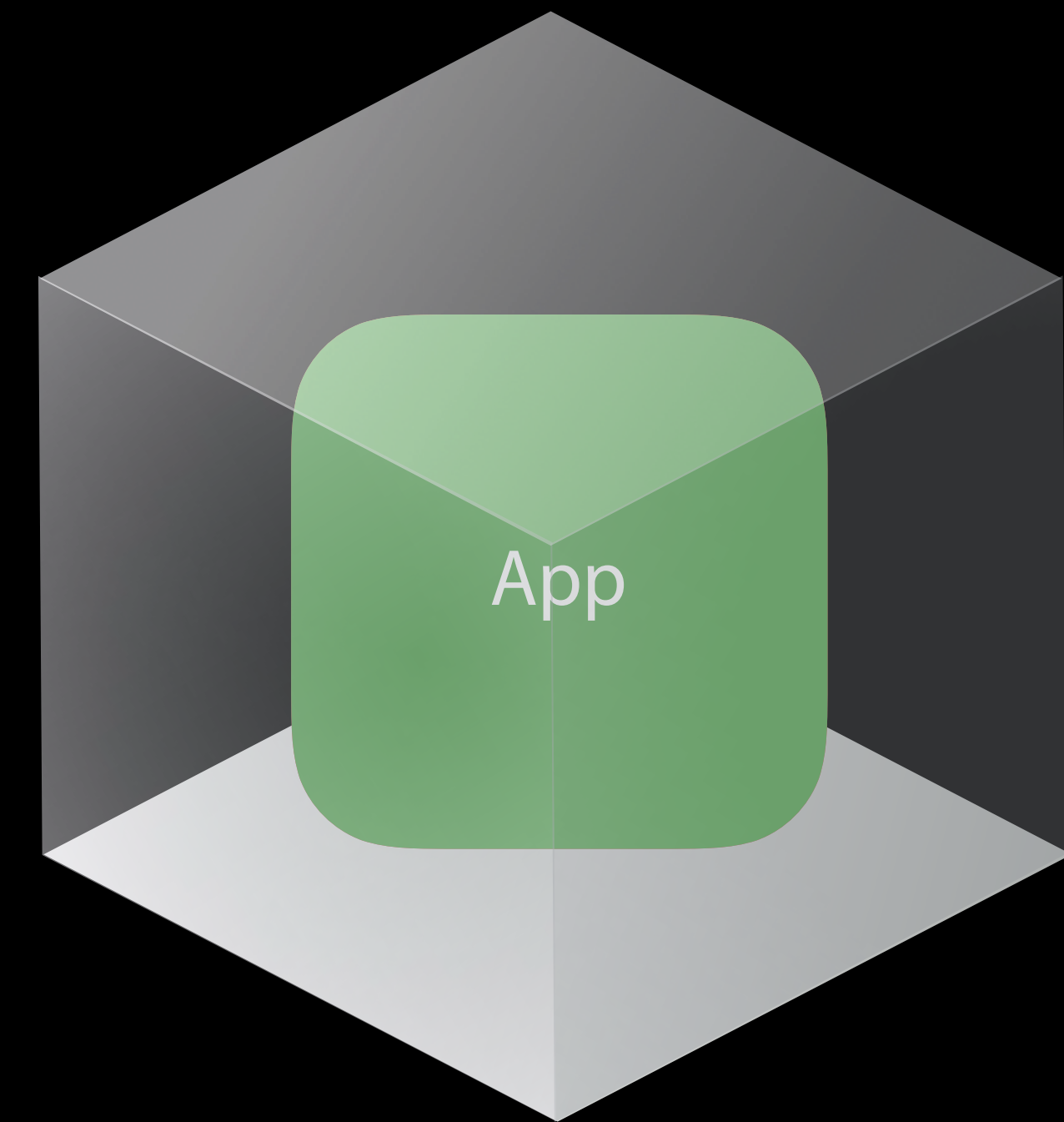1. Secure Boot
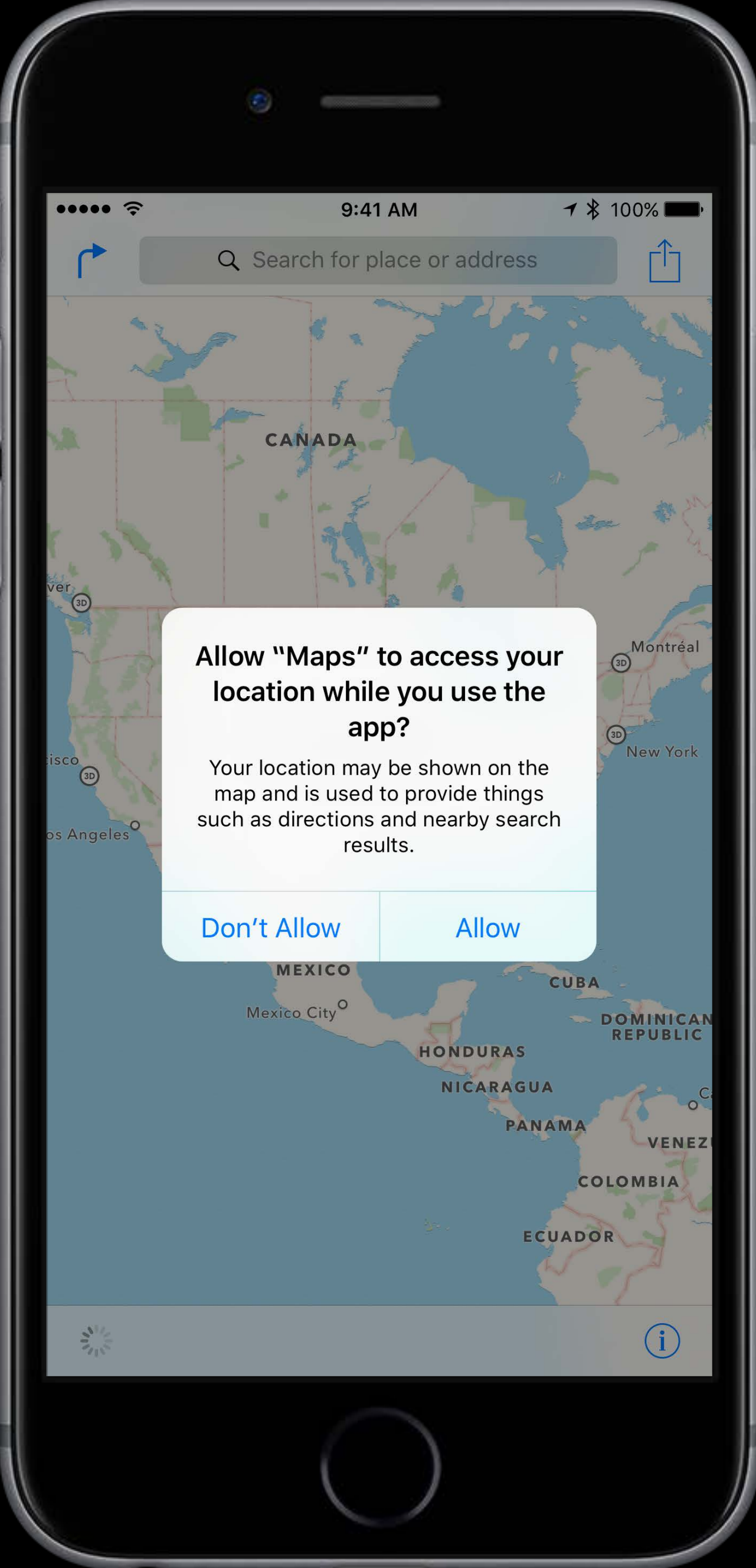2. **Data Protection**
3. Sandboxing
4. Code Signing
5. Touch ID

1 Secure Boot

2 Data Protection
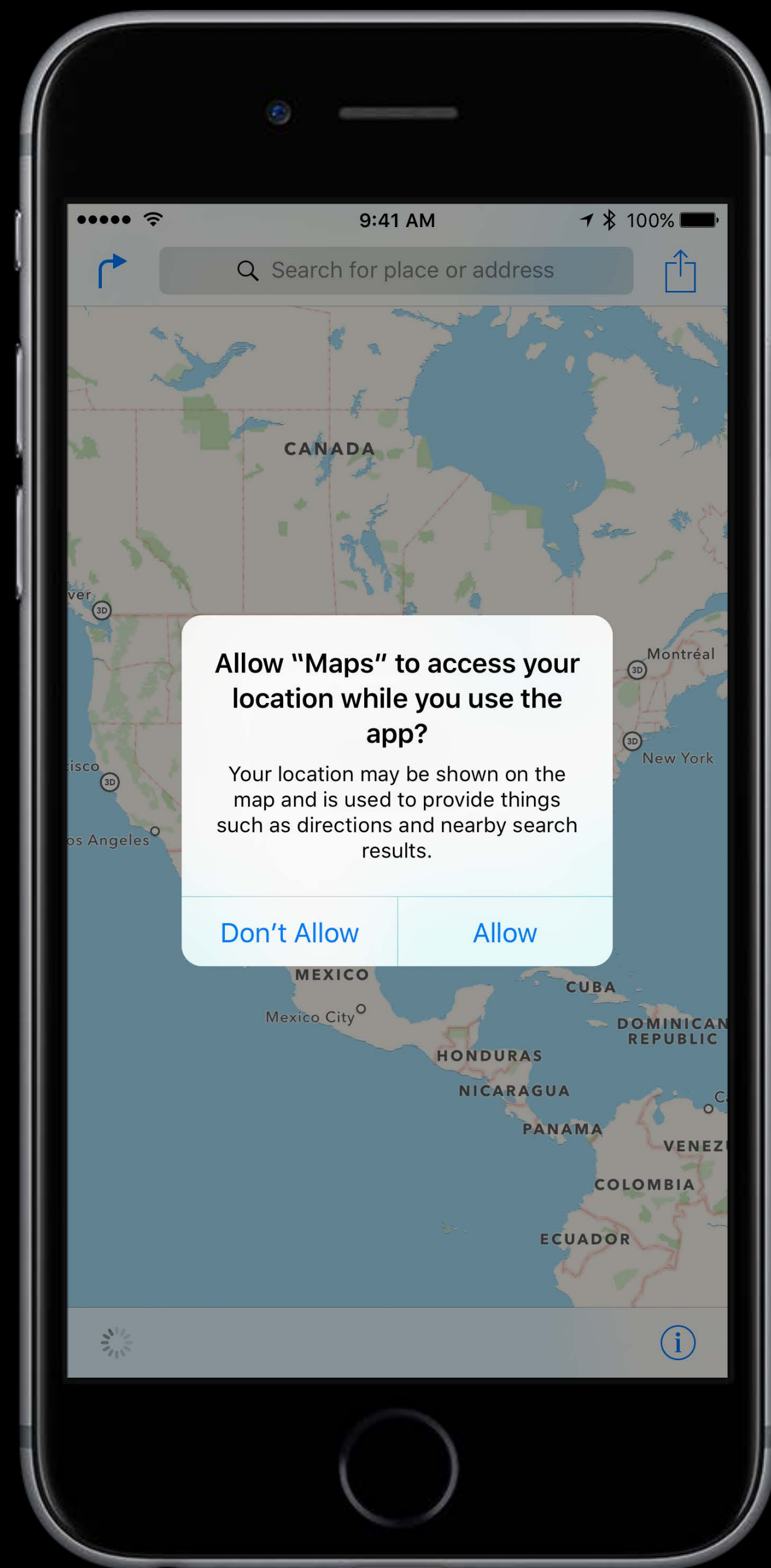
3 **Sandboxing**

4 Code Signing

5 Touch ID

Isolating data between apps

Transparency

Consent

Control

① Secure Boot

② Data Protection

③ **Sandboxing**

④ Code Signing

⑤ Touch ID

1 Secure Boot

2 Data Protection

3 Sandboxing

4 Code Signing

5 Touch ID

Code Signing

Code Signing

Attacker's first step: code execution

iOS code signing covers not just the OS, but
every app that runs

1. Secure Boot
2. Data Protection
3. Sandboxing
4. **Code Signing**
5. Touch ID

1. Secure Boot

2. Data Protection

3. Sandboxing

4. Code Signing

5. Touch ID

80

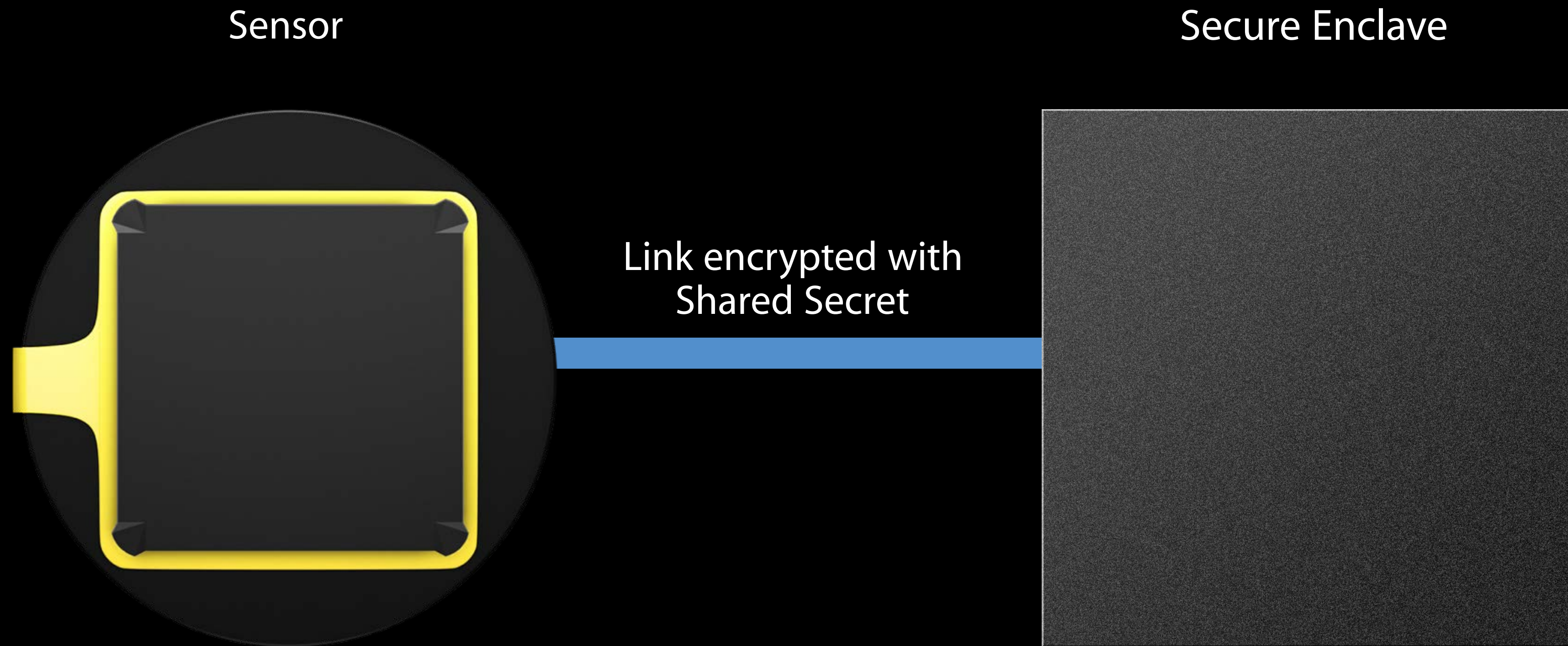Average user unlocks per day

Easy
Fast
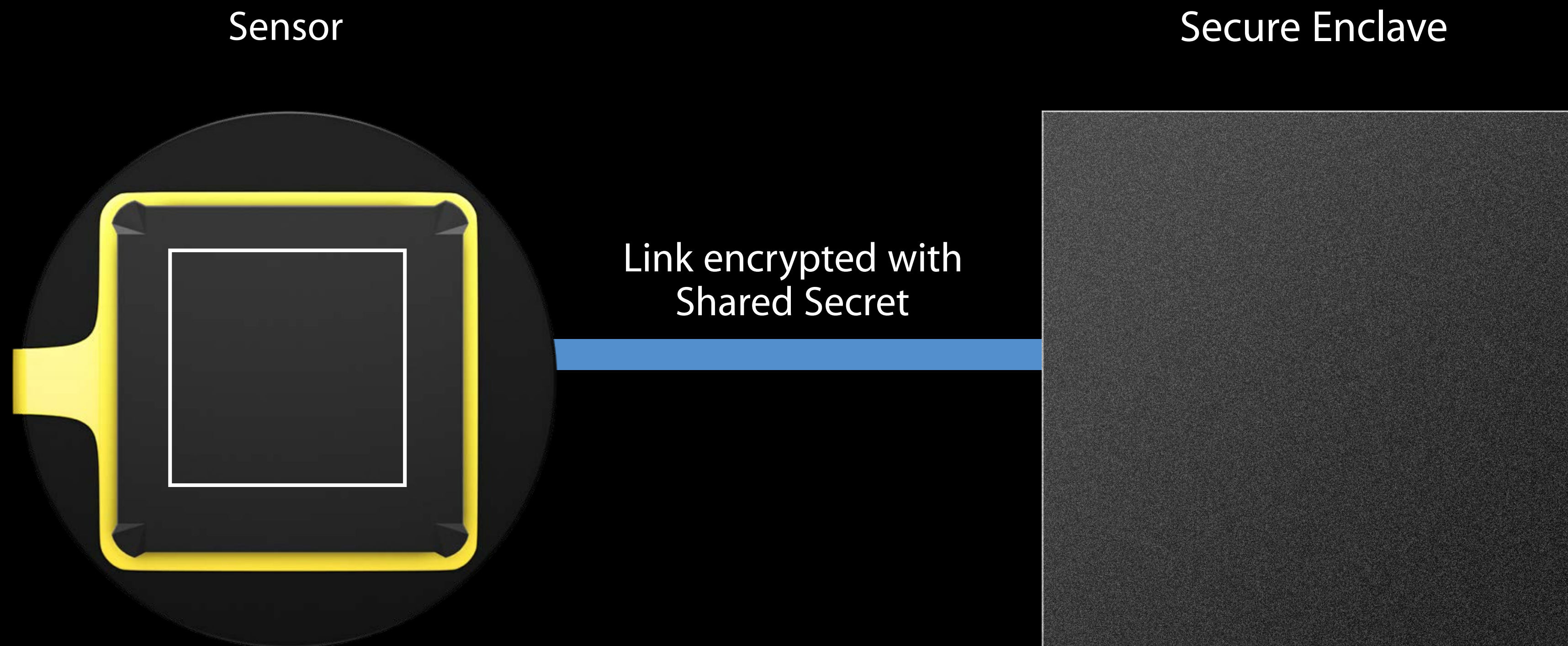Protects user data

Touch ID

Sensor

Secure Enclave

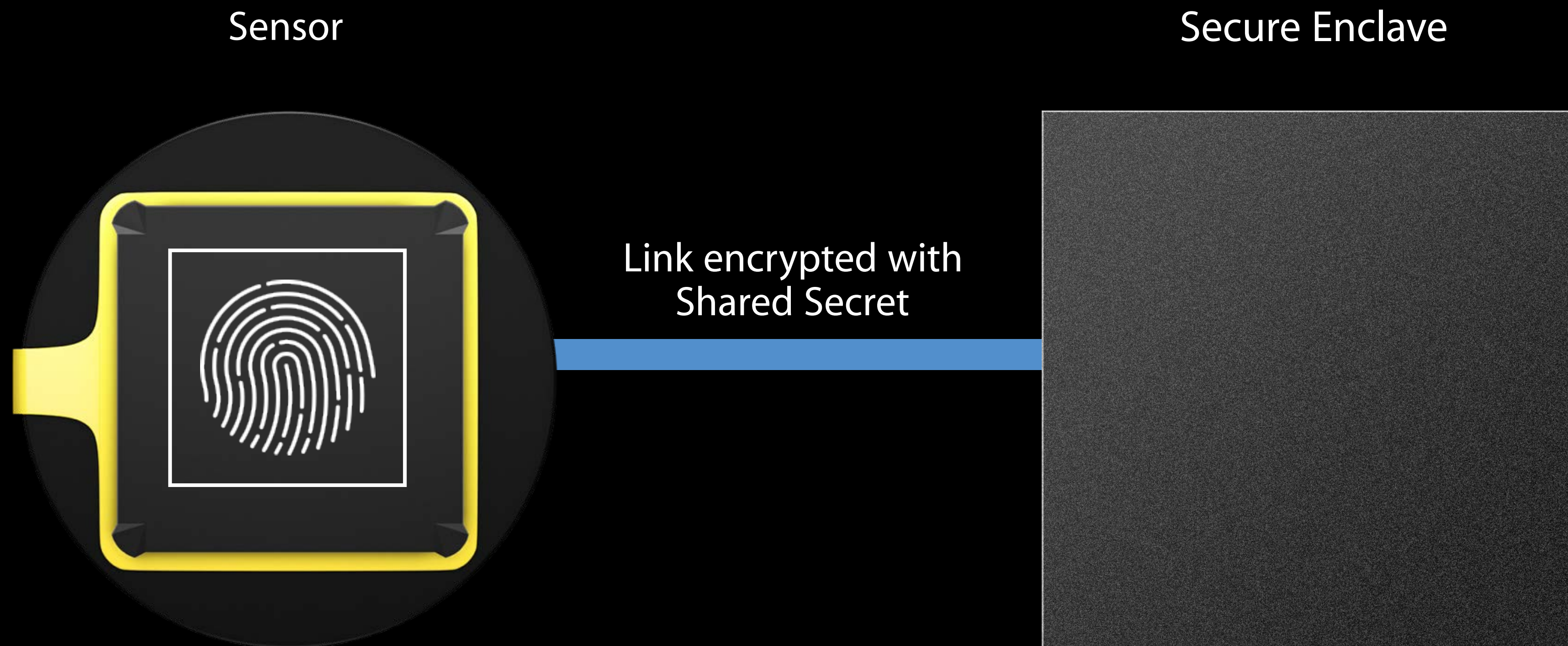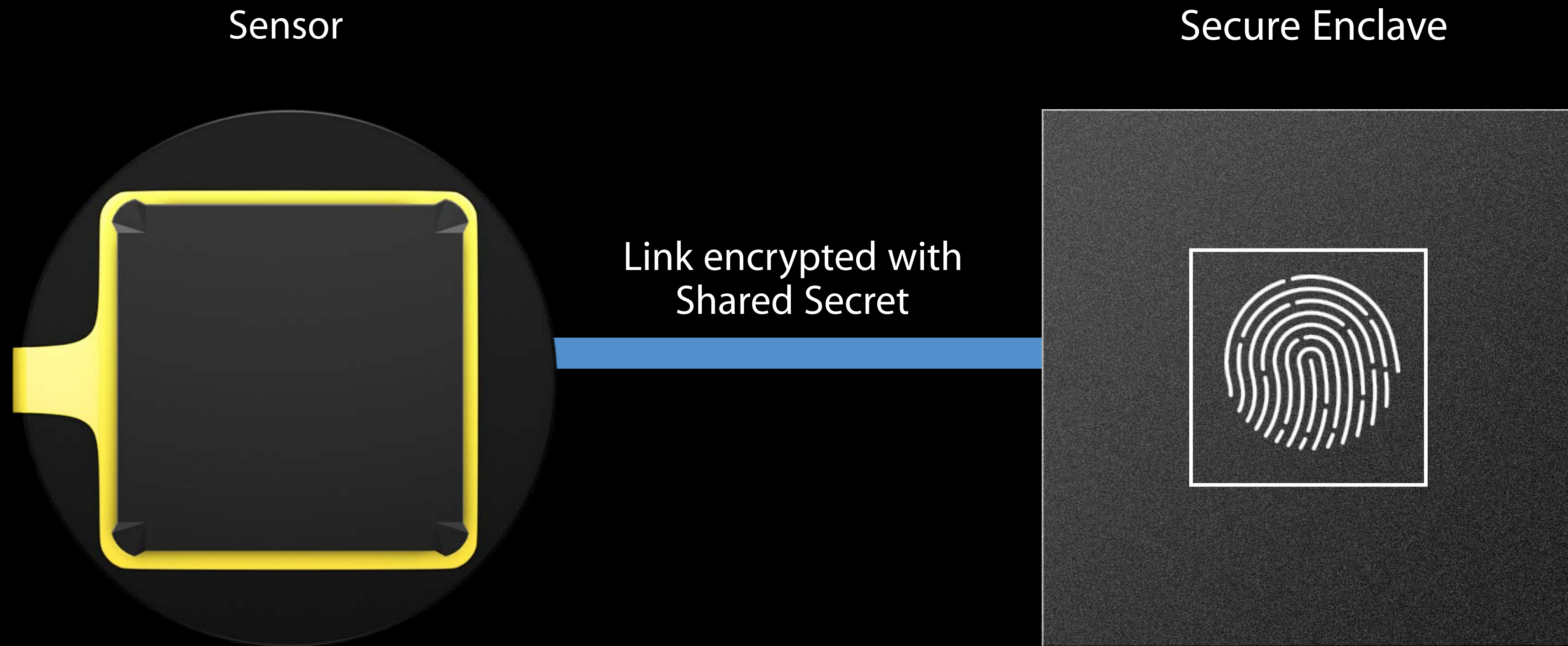# Touch ID

Sensor   Secure Enclave

Link encrypted with
Shared Secret

Touch ID

Sensor

Secure Enclave

Link encrypted with
Shared Secret

Touch ID

Sensor

Secure Enclave

Link encrypted with
Shared Secret

Touch ID

Sensor

Secure Enclave

Link encrypted with
Shared Secret

Touch ID

Sensor

Secure Enclave

Link encrypted with
Shared Secret

01001010

10100101

01010001

11100010

Touch ID

Sensor

Secure Enclave

Link encrypted with
Shared Secret

01 01 10

10 0 101

010   0 1

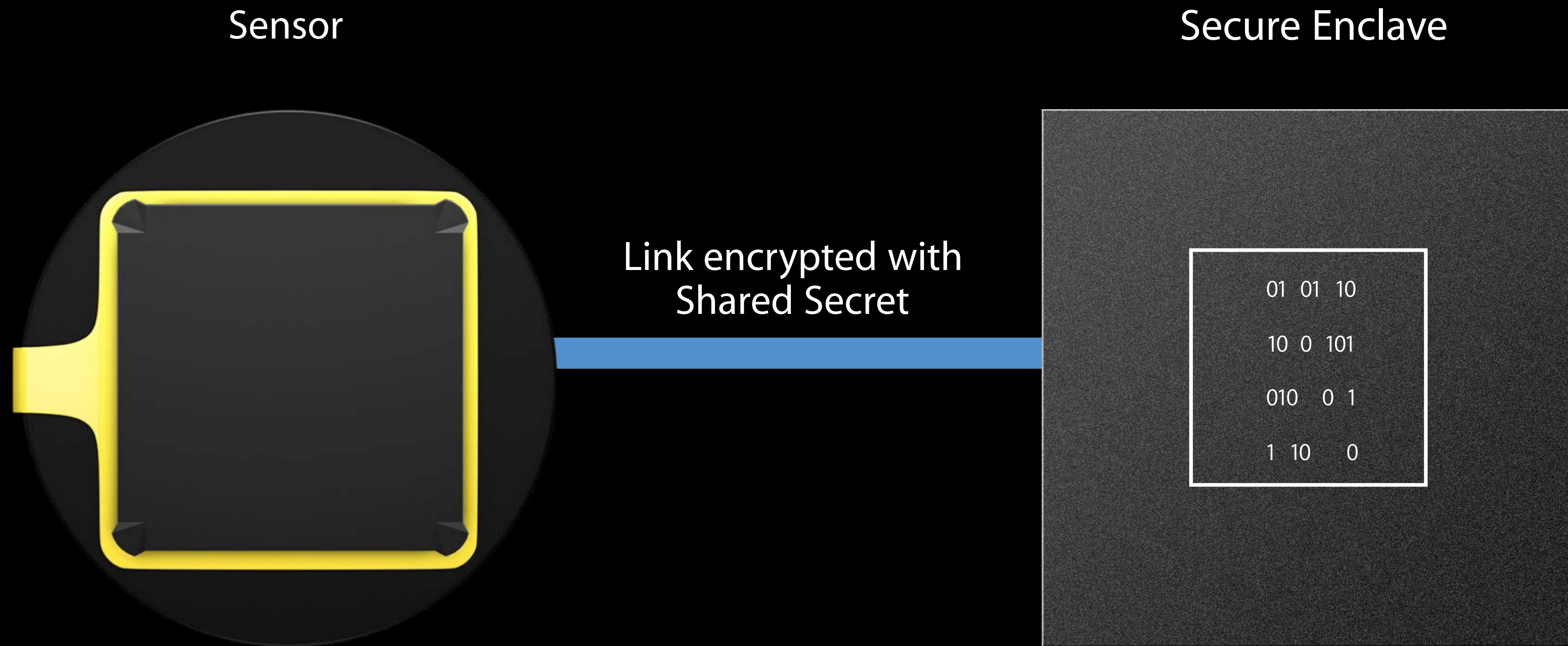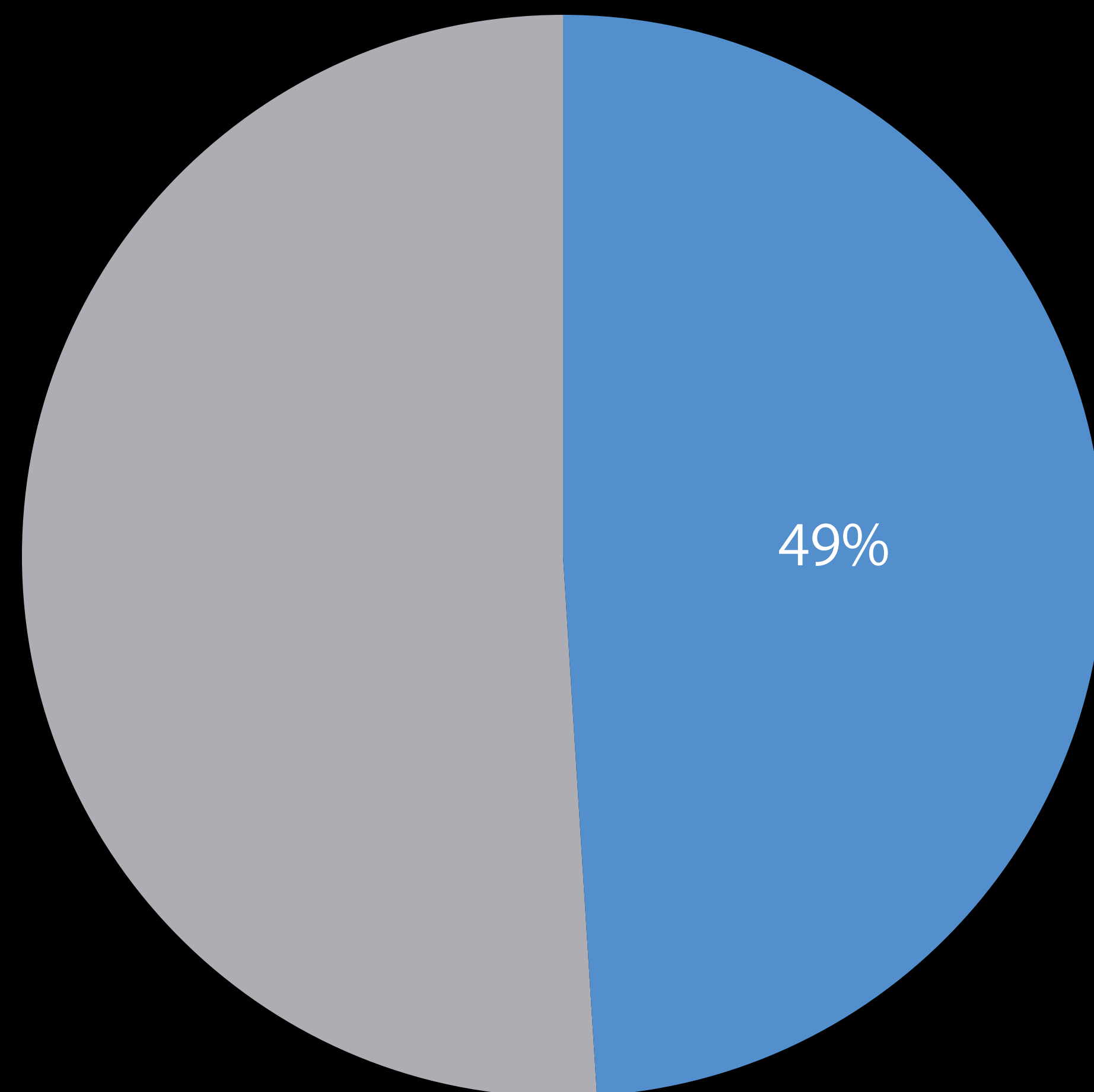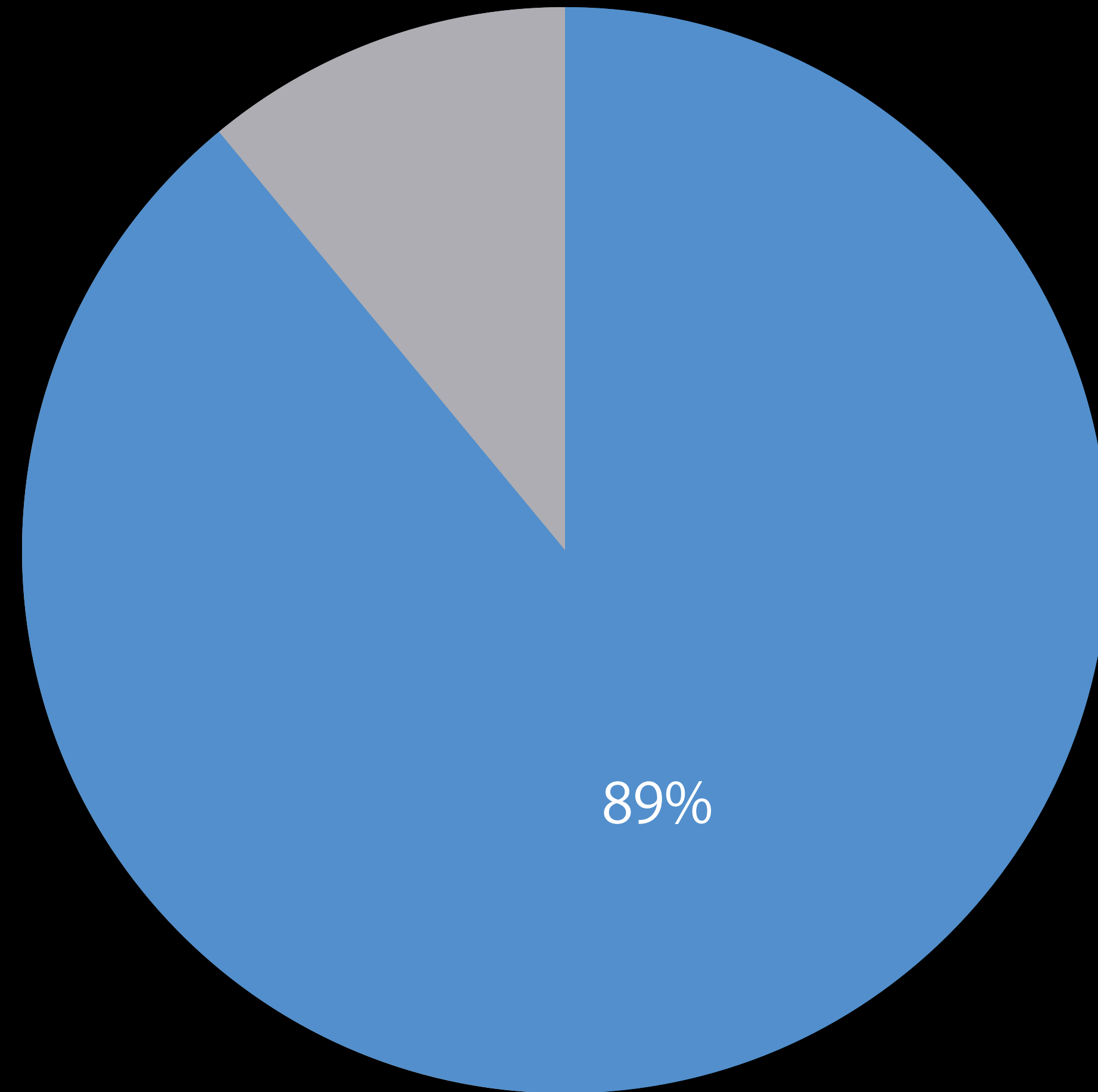1 10   0

Passcode usage before Touch ID

49%

Passcode usage after Touch ID

89%

iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps
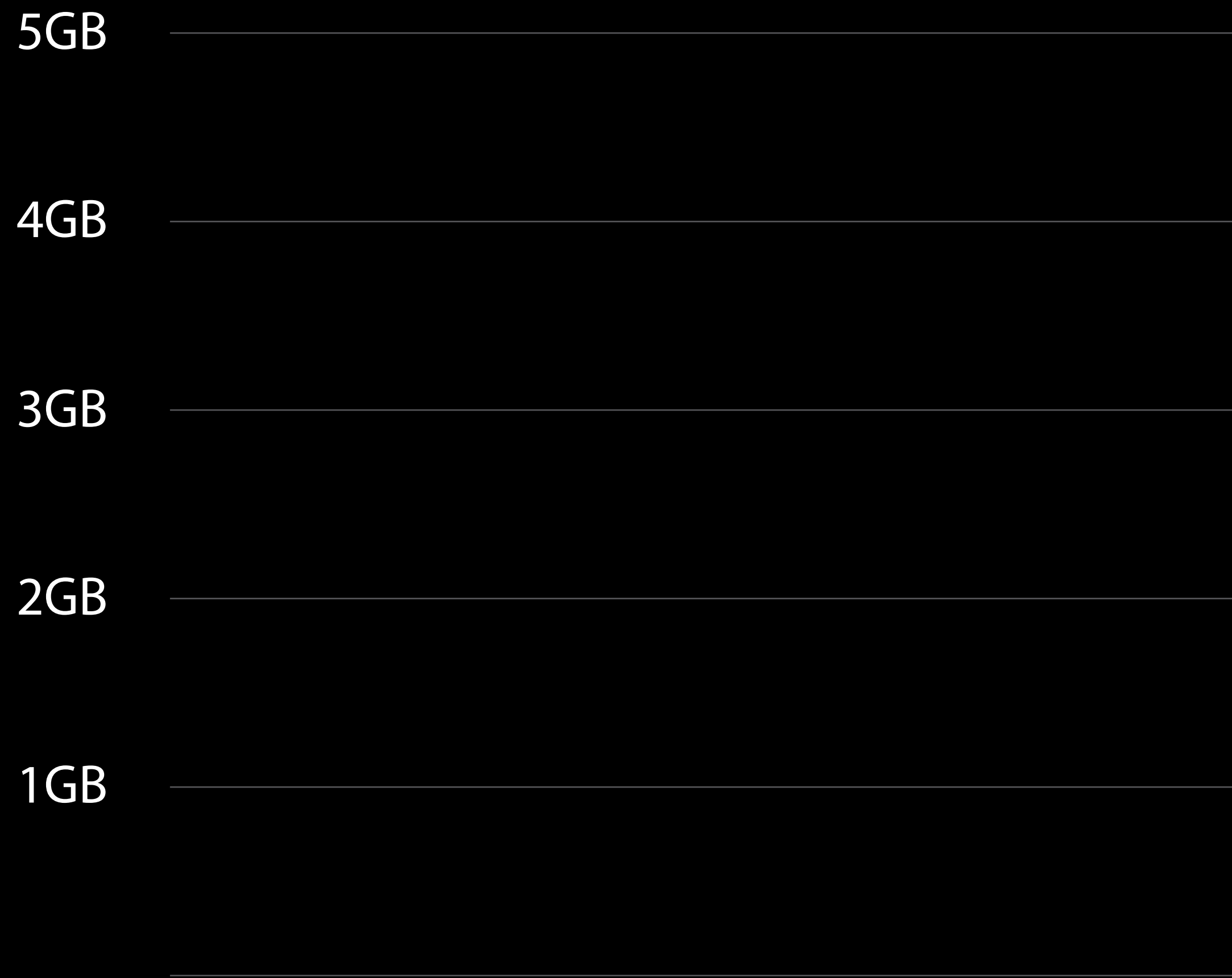
iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

Smaller install footprint

5GB ————————————————

4GB ————————————————

3GB ————————————————

2GB ————————————————

1GB ————————————————

————————————————

Smaller install footprint

5GB		
4GB		
3GB		
2GB		
1GB		
	iOS 8	iOS 9

Smaller install footprint

	4.6GB	
5GB		
4GB		
3GB		
2GB		
1GB		
	iOS 8	iOS 9

Smaller install footprint
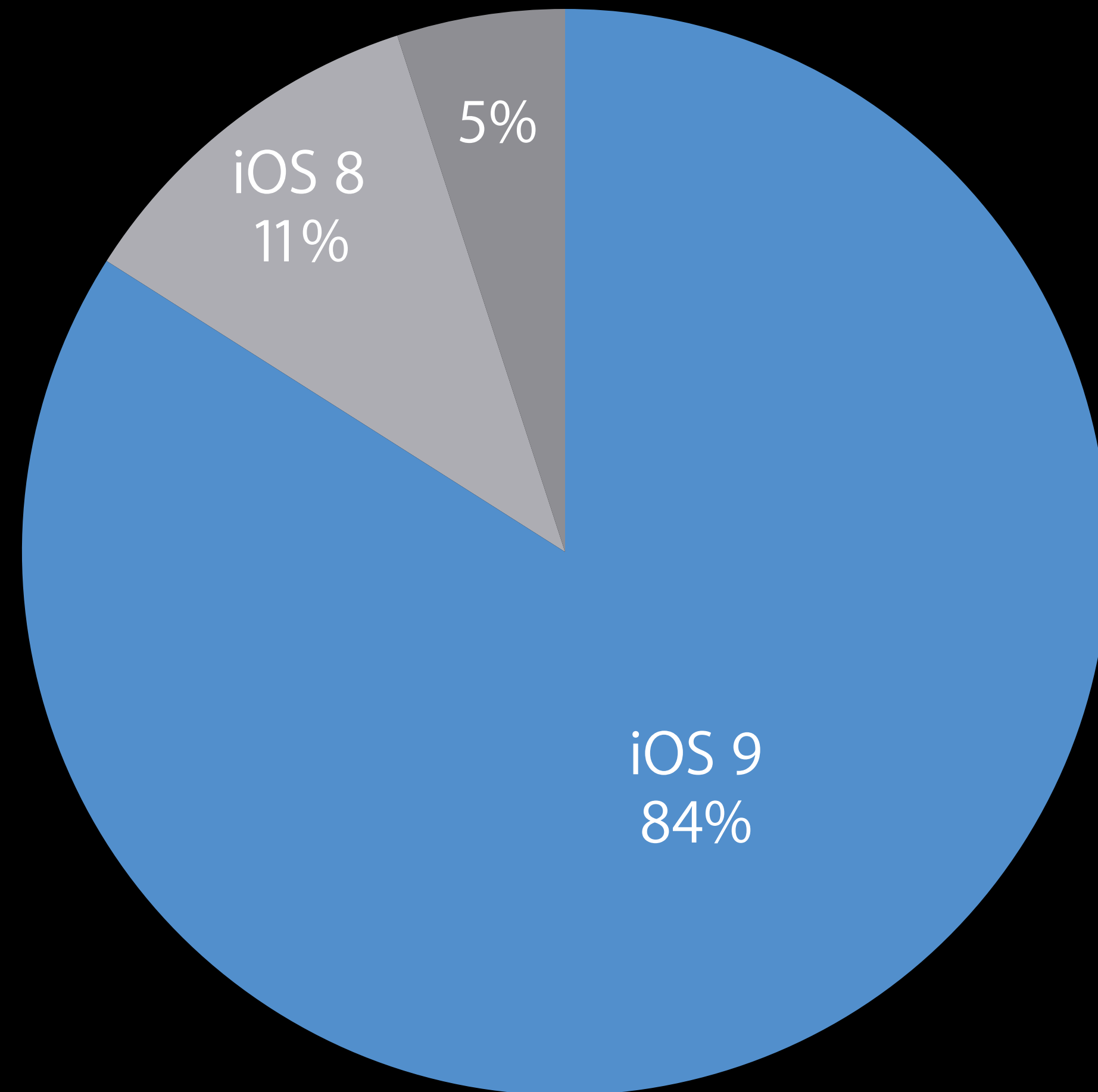
iOS Installed Base

Android Installed Base
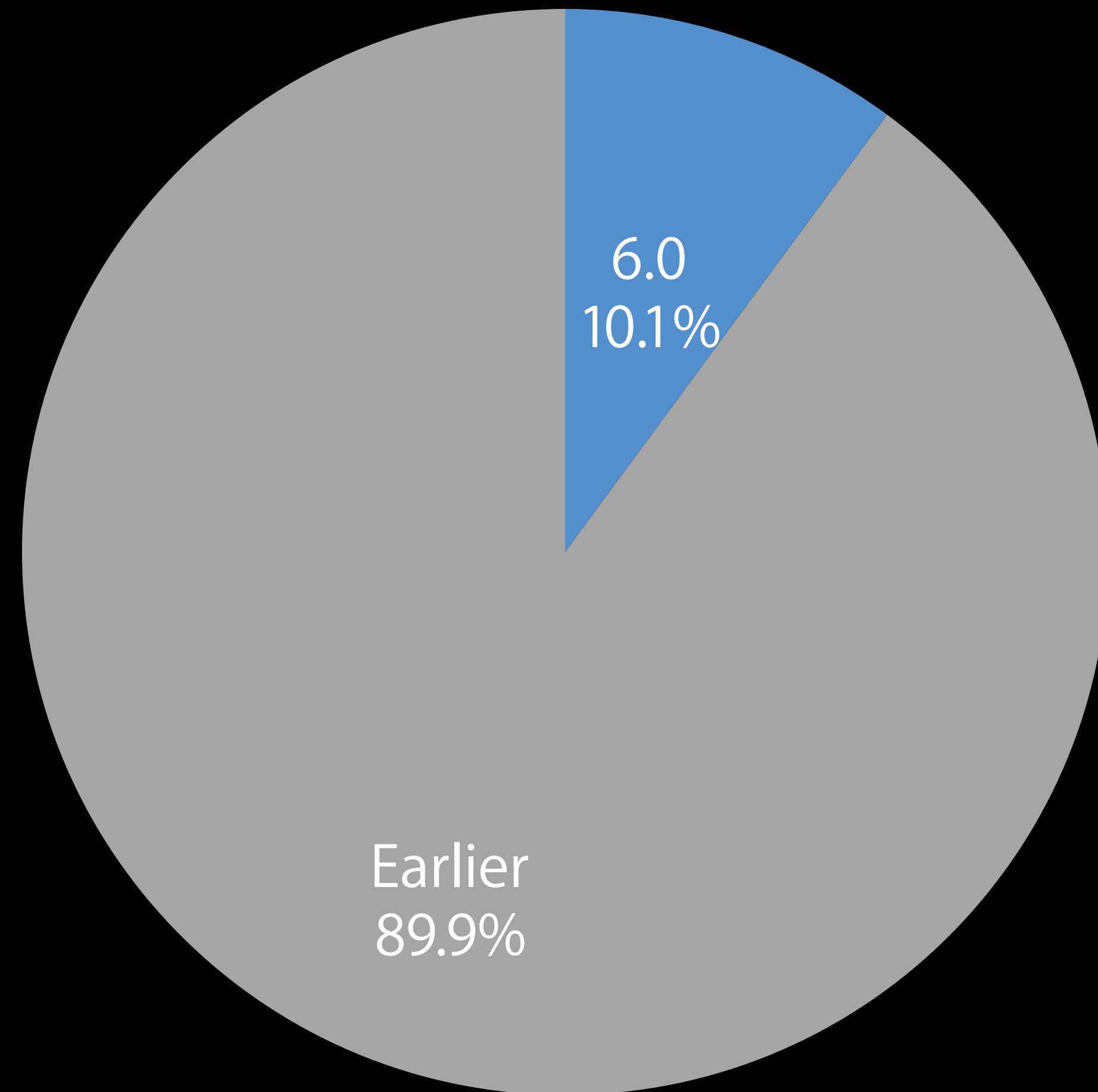
iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

iOS Platform Security

Users Upgrading their Software
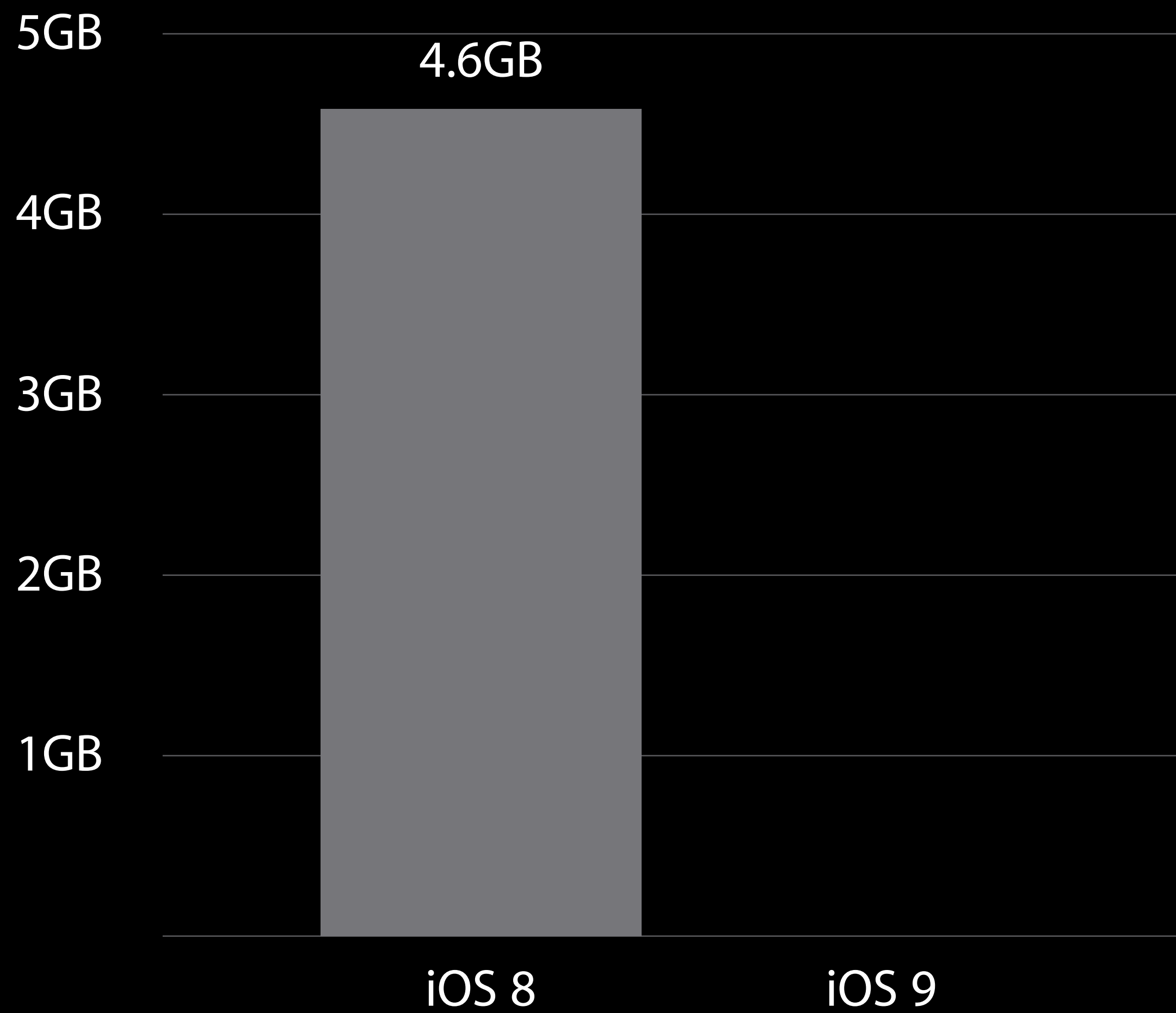
Developers Building Secure Apps

Follow Best Practices
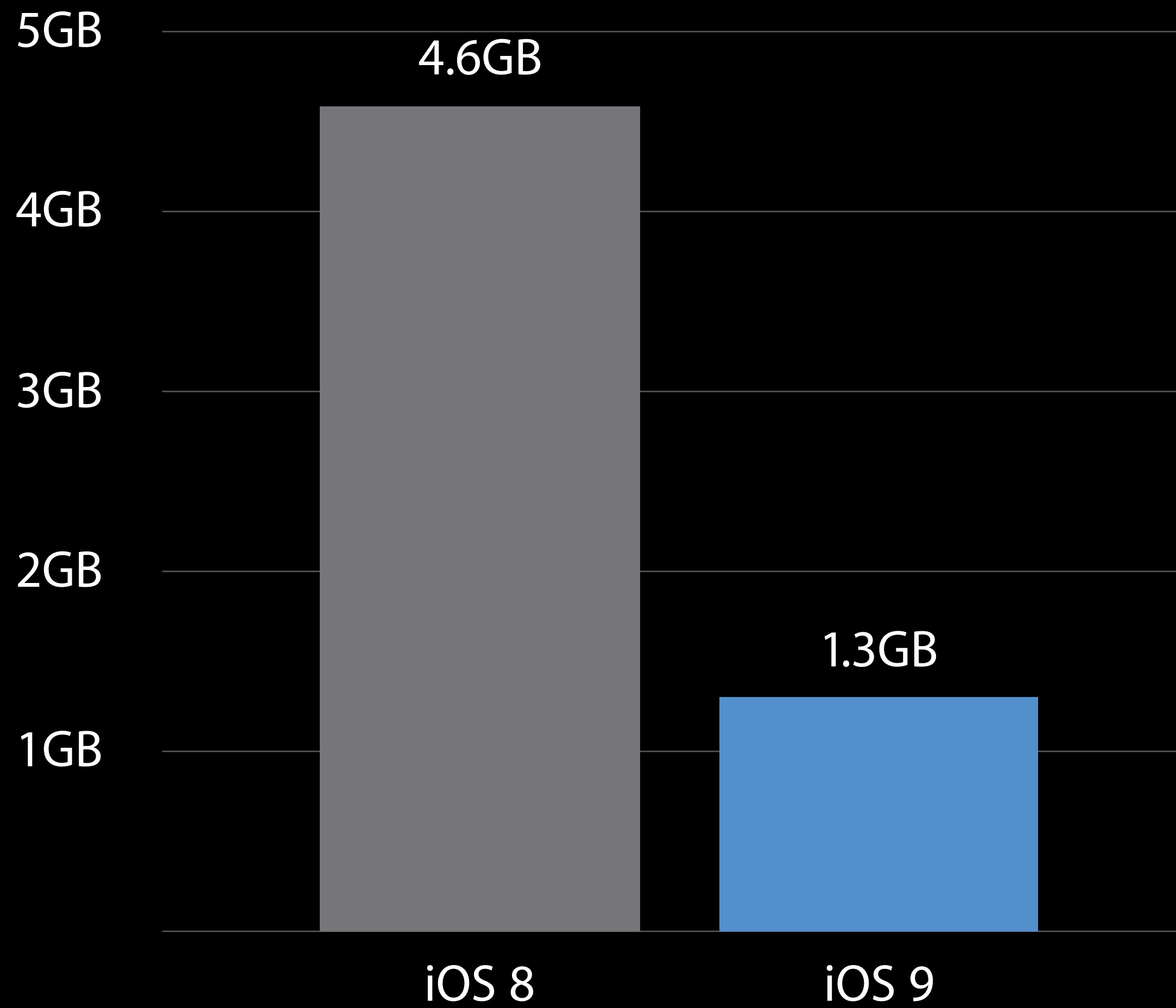
Follow Best Practices

Touch ID

App Transport Security

Follow Best Practices

Touch ID

App Transport Security

- Required by App Store at end of 2016

Follow Best Practices

Touch ID

App Transport Security

- Required by App Store at end of 2016

- TLS v1.2, with exceptions for already-encrypted bulk data like media streaming

Know Your Code

Know Your Code

You are responsible for third-party
code you include in your apps

Know Your Code

You are responsible for third-party code you include in your apps

Libraries you use may undermine app security

Know Your Code

You are responsible for third-party code you include in your apps

Libraries you use may undermine app security

Keep them current!

iOS Platform Security

Users Upgrading their Software

Developers Building Secure Apps

iOS Platform Security

Users Upgrading their Software
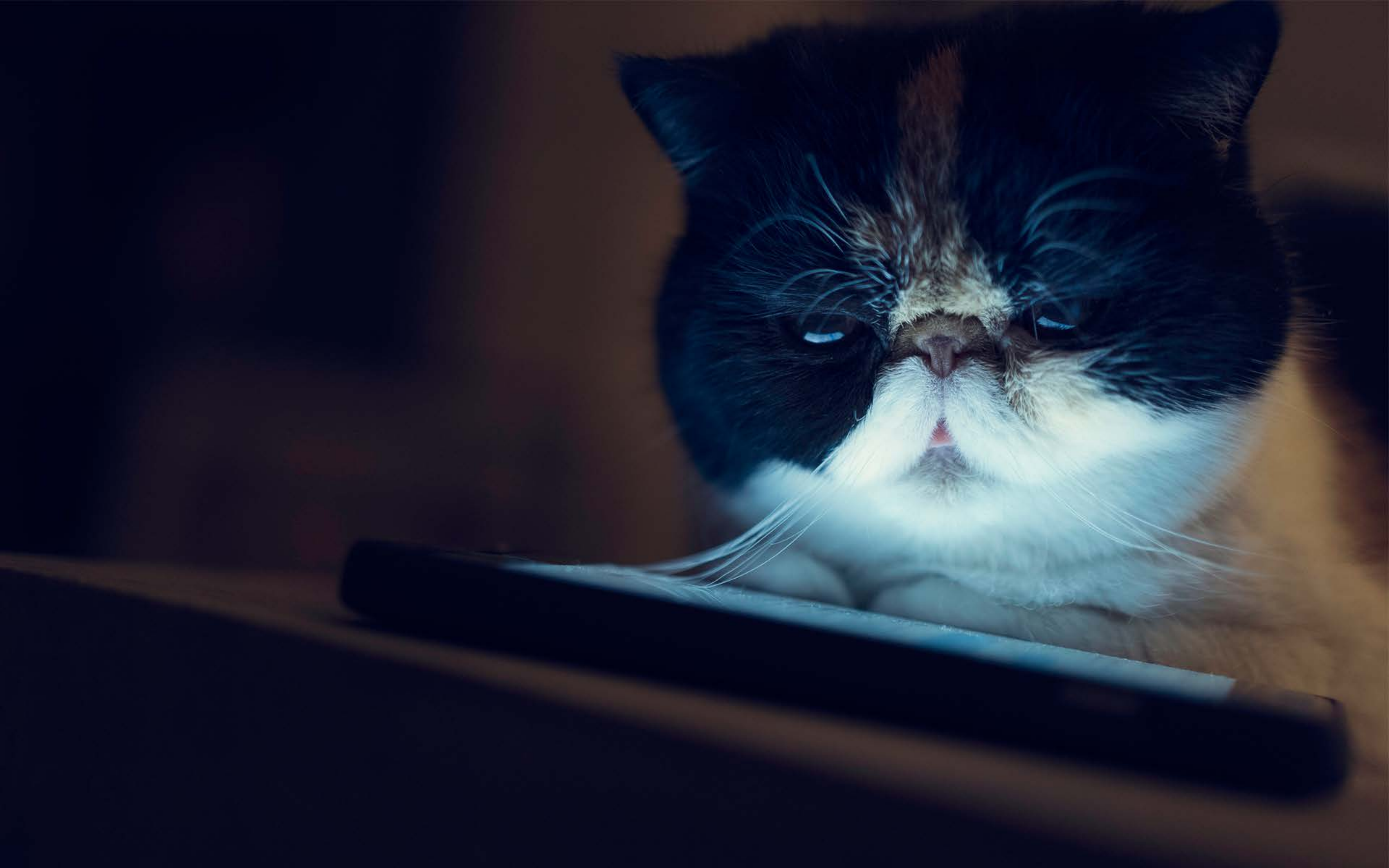
Developers Building Secure Apps

? How well are we doing?

No iOS malware at scale

5-10 vulnerabilities

$1 million

Security is a process, not a destination

More Information

https://developer.apple.com/wwdc16/705

Related Sessions

What's New in Security	Nob Hill	Tuesday 5:00PM
Engineering Privacy for Your Users	Pacific Heights	Wednesday 4:00PM

Labs

Security & Privacy Lab 1	Frameworks Lab C	Wednesday 9:00AM
Security & Privacy Lab 2	Frameworks Lab B	Thursday 9:00AM