# Apple Inc.

## Certification Practice Statement
## Worldwide Developer Relations

Version 1.27

Effective Date: June 15, 2022

# Table of Contents

# 1. INTRODUCTION

This Certification Practice Statement ("CPS") describes the practices employed by the Apple Worldwide Developer Relations Sub-CA , Apple Worldwide Developer Relations - G2 Sub-CA, Apple Worldwide Developer Relations - G3 Sub-CA, Apple Worldwide Developer Relations - G4 Sub-CA, Apple Worldwide Developer Relations - G5 Sub-CA, Apple Worldwide Developer Relations - G6 Sub-CA, Developer Authentication Sub-CA, collectively named as the "WWDR Sub-CAs" in issuing and managing digital certificates and related services. These practices, and the structure of this document, are designed to align to the requirements defined in the Apple Certificate Policy ("CP"). Where the CP defines policies that all applicable Apple Sub-CA's are required to follow, this CPS provides more detailed information about the practices employed by the WWDR Sub-CAs relating to certificate lifecycle services, such as issuance, management, revocation, renewal, and rekeying, as well as details relating to other business, legal, and technical matters specific to the WWDR Sub-CAs, collectively referred to as the WWDR Public Key Infrastructure ("WWDR PKI").

Apple Inc. ("Apple") established the Apple Root Certification Authority ("Apple Root CA") and the Apple PKI in support of the generation, issuance, distribution, revocation, administration and management of public/private cryptographic keys that are contained in CA-signed X.509 Certificates. The Apple PKI is intended to support internal and external Apple cryptographic requirements, where authentication of an organization or individual presenting a digitally signed or encrypted object to a Relying Party is of benefit to participants in the Apple PKI.

## *1.1. TRADEMARKS*

Apple, Mac, macOS, OS X, iOS, watchOS, tvOS, iPhone, iPad, iPod, iPod touch, Apple Watch and Apple TV are trademarks of Apple Inc., in the United States and other countries.

## *1.2. TABLE OF ACRONYMS*

Please refer to the CP for a table of acronyms used within this document.

## *1.3. DEFINITIONS*

For the purposes of this CPS:

- "Developer" means an individual or organization, or an individual authorized to act on behalf of an individual or company (principal), that has registered as a member of the Developer Program with WWDR and has received a Certificate from the WWDR Sub-CAs.
- "Developer Program" refers to the Apple Developer Program or the Apple Developer Enterprise Program.

- "Apple Developer website" refers to the web environment located at developer.apple.com/.
- "APNs" refers to the Apple Push Notification service, the Apple service that allows for the propagation of information to Mac, iOS, Apple Watch, and/or Apple TV devices.
- "Cloud Managed" refers to Certificates created and stored by Apple on behalf of an individual or company in the Developer program. The keys for these Certificates are generated and encrypted in Apple managed High Security Module (HSM). Eligible Developers can request content to be signed with the private key however, the private key cannot be extracted or copied. The Cloud Managed Certificates have an additional non-critical custom extension OID 1.2.840.113635.100.6.1.32.

Please refer to the CP for all other definitions used within this document.

# 2. GENERAL BUSINESS PRACTICES

This section establishes and sets forth the general business practices of the WWDR Sub-CAs.

## *2.1. IDENTIFICATION*

The practices set forth in this CPS apply exclusively to the WWDR Sub-CAs. This CPS is structured similarly to the CP, disclosing details of the practices employed by the WWDR Sub-CAs that address the more general requirements defined in the CP. This document assumes the reader is familiar with the general concepts of digital signatures, certificates, and public-key infrastructure. If the reader is new to Public Key Infrastructure concepts, the reader may choose to consult the Introduction and Overview sections of the Trust Service Principles and Criteria for Certification Authorities, a document published by the Chartered Professional Accountants of Canada (“CPA Canada”) and freely available for download from their web site, www.webtrust.org. The document contains an overview of PKI, including an orientation on key concepts such as digital signatures, asymmetric key pairs, certification authorities, registration authorities, policy and practice statements, and business issues and considerations.

For the purposes of this CPS, the term Apple PKI refers collectively to Apple PKI Service Providers and End Entities. Apple PKI Service Providers consist of (1) Apple Certification Authorities ("CAs"), including the Apple Root CAs, the WWDR Sub-CAs and their related management teams that generate, issue, distribute, revoke and manage cryptographic keys and Certificates, (2) Apple Registration Authorities ("Apple RA"), and (3) the Apple CA Policy Authority ("Apple PA," or “PA”). End Entities are Subscribers of Certificates.

The WWDR Sub-CAs issues and administer Certificates in accordance with policies in the Apple CP document.

## *2.2. COMMUNITY AND APPLICABILITY*

This CPS is applicable to the following Certificates and Cloud Managed Certificate versions indicated by an (*) issued by the WWDR Sub-CAs:

- WWDR iOS Software Development Certificates (“iOS Development Certificates”)*
- WWDR iOS Software Submission Certificates (“iOS Submission Certificates”)*
- WWDR Apple Push Notification service Development SSL Certificates (“Development SSL Certificates”)

- WWDR Apple Push Notification service Production SSL Certificates (“Production SSL Certificates”)
- WWDR Push Certificate Signing Request Signing Certificates (“Push CSR Signing Certificates”)
- WWDR Safari Extension Signing Certificates (“Safari Certificates”)
- WWDR Mac App Development Certificates (“Mac App Development Certificates”)*
- WWDR Mac App Submission Certificates (“Mac App Submission Certificates”)*
- WWDR Mac Installer Package Submission Certificates (“Mac Installer Package Submission Certificates”)*
- Mac App Store Application Signing Certificates (“Mac App Store Application Certificates”)
- Mac App Store Installer Package Signing Certificates (“Mac App Store Installer Package Certificates”)
- Mac App Store Receipt Signing Certificates
- Mac Provisioning Profile Signing Certificates
- Pass Certificates
- Website Push Notification Certificates
- OS X Server Authentication Certificates
- VoIP Services Push Certificates
- Apple Pay Merchant Certificates
- Apple Pay Pass Certificates
- TestFlight Distribution Certificates
- WatchKit Services Certificates
- Apple Pay Provisioning Encryption Certificates
- Enhanced Pass Certificates
- tvOS Application Signing Certificates

- WWDR Apple Push Services Client Authentication G2 Certificates
- Apple Pay Merchant Client Authentication Certificates
- WWDR Apple Development Signing Certificates ("Apple Development Certificates")*
- WWDR Apple Distribution Signing Certificates ("Apple Distribution Certificates")*
- Swift Package Collection Signing Certificates
- Order Type ID Signing Certificates
- Identity Processing Certificates

* Cloud Managed versions of this certificate are issued and stored as described in section 3.5 and 3.6. Cloud Managed certificates include an additional non-critical custom extension OID 1.2.840.113635.100.6.1.32 to indicate they are Cloud managed.

Certificates used exclusively for functions internal to Apple Products and/or Apple processes, such as device profile signing Certificates, event log signing Certificates, etc., are not included within the scope of this CPS.

### 2.2.1. iOS Development Certificates

The WWDR Sub-CA issues and administers Certificates that may be used by Developers to digitally sign a software application, enabling the application to be tested on an Apple iPhone, iPad, iPod touch, Apple Watch and/or Apple TV.

### 2.2.2. iOS Submission Certificates

The WWDR Sub-CA issues and administers Certificates used by a Developer to digitally sign software applications for submission to Apple, or for distribution to their internal employees.

### 2.2.3. Development Client SSL Certificates

The WWDR Sub-CA issues and administers Client SSL Certificates that are used by Developers to provide SSL connectivity and client authentication for the Apple Push Notification service Development environment.

### 2.2.4. Production Client SSL Certificates

The WWDR Sub-CA issues and administers Client SSL Certificates that are used by Developers to provide SSL connectivity and client authentication for the Apple Push Notification service Production environment.

### 2.2.5. Push CSR Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used to digitally sign Certificate Signing Requests, enabling these Certificate Signing Requests to be submitted to the Apple Push Certificate Portal.

### 2.2.6. Safari Extension Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign a Safari web browser extension, enabling it to be installed in the Safari application.

### 2.2.7. Mac Application Development Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign a Mac application bundle, enabling it to be tested on an Apple Mac.

### 2.2.8. Mac Application Submission Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign a Mac application bundle, enabling it to be submitted to Apple.

### 2.2.9. Mac Installer Package Submission Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign a Mac installer package, enabling it to be submitted to Apple.

### 2.2.10. Mac App Store Application Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple to sign application bundles distributed through the Mac App Store.

### 2.2.11. Mac App Store Installer Package Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple to sign installer packages distributed through the Mac App Store.

### 2.2.12. Mac App Store Receipt Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple to sign receipts for applications delivered through the Mac App Store.

### 2.2.13.Mac Provisioning Profile Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple to sign provisioning profiles for Mac application development and submission to the Mac App Store.

### 2.2.14.Pass Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign passes for Apple Wallet, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates

### 2.2.15.Website Push Notification Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign notification packages for websites on OS X, enabling them to be authenticated on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.

### 2.2.16.OS X Server Authentication Certificates

The Developer Authentication Sub-CA issues and administers Certificates that are used by Developers to access WWDR API's through Developer Tools.

### 2.2.17. VoIP Services Push Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers of VoIP apps to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates for VoIP apps.

### 2.2.18.Apple Pay Merchant Certificates

The WWDR-G2 Sub-CA issues and administers Certificates that are used by Apple to encrypt transaction data sent from Apple to a merchant developer.

### 2.2.19.Apple Pay Pass Certificates

The WWDR-G2 Sub-CA issues and administers Certificates that are used by Apple to digitally sign Apple Wallet passes, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.

### 2.2.20.TestFlight Distribution Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple to digitally sign TestFlight apps, and third-party macOS software packages, enabling them to be installed on an Apple device.

### 2.2.21.WatchKit Services Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to create and maintain SSL connectivity to the Apple Push Notification service, allowing them to push updates to watch complications.

### 2.2.22.Apple Pay Provisioning Encryption Certificates

The WWDR-G2 Sub-CA issues and administers Certificates that are used by Apple Pay participating issuers to encrypt data during the provisioning process.

### 2.2.23.Enhanced Pass Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to digitally sign enhanced passes for Apple Wallet that contain an NFC dictionary, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.

### 2.2.24.tvOS Application Signing Certificates

The WWDR-G2 Sub-CA issues and administers Certificates that are used by Apple to sign tvOS application bundles distributed through the App Store on Apple TV.

### 2.2.25.WWDR Apple Push Services Client Authentication G2 Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to provide SSL connectivity and client authentication for the Apple Push Notification service Development and Production environments.

### 2.2.26.Apple Pay Merchant Client Authentication Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Apple Pay Merchant Developers to provide SSL connectivity and client authentication to the Apple Pay environment.

### 2.2.27.Apple Development Certificates

The WWDR Sub-CA issues and administers Certificates that may be used by Developers to digitally sign a software application or application bundle, enabling the application to be tested on an Apple Mac, iPhone, iPad, iPod touch, Apple Watch and/or Apple TV.

### 2.2.28.Apple Distribution Certificates

The WWDR Sub-CA issues and administers Certificates used by a Developer to digitally sign software applications and macOS application bundles for submission to Apple, or for distribution of software applications to their internal employees.

### 2.2.29.Swift Package Collection Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used by Developers to sign Swift Package Collections.

### 2.2.30.Order Type ID Signing Certificates

The WWDR Sub-CA issues and administers Certificates that are used to digitally sign order information payloads for Apple Wallet, enabling them to be displayed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.

### 2.2.31.Identity Processing Certificates

The WWDR Sub-CA issues and administers Certificates that are used by approved Developers to decrypt identity elements.

## *2.3. CONTACT DETAILS*

The CA's Certificate Policies are administered by the Apple CA Policy Authority. The contact information for this CPS is:

Apple CA Policy Authority
C/O General Counsel
Apple Inc.
One Apple Park Way
Cupertino, CA 95014

(408) 996-1010
policy_authority@apple.com

## *2.4. APPORTIONMENT OF LIABILITY*

For Certificates issued to Developers, a subscriber agreement is incorporated in the applicable License Agreement. For Mac App Store Receipt Signing Certificates, a relying party agreement will be incorporated into the applicable License Agreement. There is not an applicable Relying Party agreement for any other Certificate from the WWDR Sub-CAs as the relying parties are internal to

Apple. Except as provided herein, parties external to Apple are expressly prohibited from placing reliance on any aspects of the WWDR PKI.

### 2.4.1. Warranties to Subscribers and Relying Parties

The WWDR Sub-CAs do not warrant the use of any Certificate to any Subscriber or Relying Party.

### 2.4.2. CA Disclaimers of Warranties

To the extent permitted by applicable law, subscriber agreements disclaim warranties from Apple, including any warranty of merchantability or fitness for a particular purpose.

### 2.4.3. CA Limitations of Liability

To the extent permitted by applicable law, subscriber agreements shall limit liability on the part of Apple and shall exclude liability for indirect, special, incidental, and consequential damages.

### 2.4.4. Subscriber Warranties

For Certificates issued to Developers, subscriber agreements shall require Subscribers to warrant that:

- They will take no action to interfere with the normal operation of a Certificate or products that rely on such certificates;
- They are solely responsible for preventing any unauthorized person from having access to the Subscriber's account and Subscriber's private key stored on any device for which the Subscriber is developing software for Apple platforms; and
- The Certificates are being used exclusively for authorized and legal purposes.

### 2.4.5. Private Key Compromise

Apple reserves the right to revoke any Certificates, without notice, if it believes the Subscriber's private key has been compromised, or upon request from the Subscriber.

### 2.4.6. Subscriber and Relying Party Liability

Subscribers and Relying Parties will hold Apple harmless from any and all liabilities, losses, actions, damages or claims (including all reasonable expenses, costs, and attorney fees) arising out of or relating to their use of any digital Certificate.

## *2.5. FINANCIAL RESPONSIBILITY*

This section sets forth policies as requirements on the WWDR Sub-CAs related to indemnification by Relying Parties and disclosure of fiduciary relationships in relying party agreements.

### 2.5.1. Indemnification by Subscribers and Relying Parties

Any subscriber or relying party agreement may, at Apple's discretion, include an indemnification clause by Subscribers and/or Relying Parties.

### 2.5.2. Fiduciary Relationships

There is no fiduciary relationship between Apple and Subscribers and/or Relying Parties.

## *2.6. INTERPRETATION AND ENFORCEMENT*

Interpretation and enforcement of any subscriber or relying party agreement is governed by the terms and conditions in the SDK License Agreement.

### 2.6.1. Governing Law

Governing law is set forth in the SDK License Agreement.

### 2.6.2. Severability, Survival, Merger, Notice

Severability, survival, merger and notice if applicable, is governed by the terms and conditions in the SDK License Agreement.

### 2.6.3. Dispute Resolution Procedures

Dispute resolution procedures are set forth in the SDK License Agreement.

## *2.7. FEES*

This section sets forth policies associated with any fees charged to Subscribers for certification authority services for each type of Certificate.

### 2.7.1. Certificate Issuance or Renewal Fees

No fees are charged for this service. Certificates issued to Developers are available at no additional cost to members of the Developer Program. Certificates are valid for the duration of the membership period unless otherwise revoked.

### 2.7.2. Certificate Access Fees

No fees are charged for this service.

### 2.7.3. Revocation or Status Information Access Fees

No fees are charged for this service.

### 2.7.4. Fees for Other Services

No other fees are charged for CA services.

### 2.7.5. Refund Policy

Not applicable.

## *2.8. PUBLICATION AND REPOSITORY*

The WWDR Sub-CAs operate a private repository, which is not publicly accessible.

### 2.8.1. Publication of CA Information

The latest version of this CPS for the WWDR Sub-CAs can be found at https://www.apple.com/certificateauthority.

### 2.8.2. Frequency of Publication

The CPS will be published in the repository after approval by the Apple PA. Sub-CA Certificates and CRLs are published in the repository after issuance.

### 2.8.3. Access Controls

There is no public repository of certificates. Developers shall have access to their own Certificates through the Apple Developer website.

## *2.9. COMPLIANCE AUDIT REQUIREMENTS*

The WWDR Sub-CAs adopt wholly all policies under this section in the CP.

## *2.10. CONDITIONS FOR APPLICABILITY*

This section sets forth practices related to the use of the WWDR Sub-CAs.

### 2.10.1. Permitted Uses

The WWDR Sub-CAs will create keys, manage keys, issue Certificates, manage key life cycles, manage certificate life cycles, operate a private repository, and perform other functions to support distribution for the following types of Certificates and Cloud Managed Certificate versions indicated with an (*) :

- iOS Development Certificates*: This type of Certificate may be used by Developers authorized by Apple to digitally sign a software application

enabling the application to be tested on an Apple Mac, iPhone, iPad, iPod touch, Apple Watch and/or Apple TV.

- iOS Submission Certificates*: This type of Certificate may be used by a Developer to digitally sign software applications for submission to Apple, or for distribution to their internal employees.
- Development SSL Certificates: This type of Certificate may be used by Developers authorized by Apple to create and maintain SSL connectivity to the Apple Push Notification service Development environment enabling remote notifications to be sent via the Apple Push Notification service Development server to an Apple Mac, iPhone, iPad, iPod touch, Apple Watch and/or Apple TV.
- Production SSL Certificates: This type of Certificate may be used by Developers authorized by Apple to create and maintain SSL connectivity to the Apple Push Notification service Production environment enabling remote notifications to be sent via the Apple Push Notification service Production server to an Apple Mac, iPhone, iPad, iPod touch, Apple Watch and/or Apple TV.
- Push CSR Signing Certificates: This type of Certificate, may be used by Apple authorized Developers, enabling CSRs to be digitally signed and submitted to the Apple Push Certificate Portal.
- Safari Certificates: This type of Certificate may be used by Developers authorized by Apple to digitally sign a Safari web browser extension, enabling it to be installed by Safari users.
- Mac Application Development Certificates*: This type of Certificate may be used by Developers authorized to sign a macOS application bundle enabling it to be tested on a Mac used for application development.
- Mac Application Submission Certificates*: This type of Certificate may be used by Developers authorized to sign a macOS application bundle enabling it to be submitted to the Mac App Store.
- Mac Installer Package Certificates*: This type of Certificate may be used by Developers authorized to sign a macOS installer package enabling it to be submitted to the Mac App Store.
- Mac App Store Application Certificates: This type of Certificate may be used by Apple to sign a macOS application bundle enabling it to be distributed via the Mac App Store.

- Mac App Store Installer Package Certificates: This type of Certificate may be used by Apple to sign a macOS installer package enabling it to be distributed via the Mac App Store.
- Mac App Store Receipt Signing Certificates: This type of Certificate may be used by Apple to sign receipts for applications delivered through the Mac App Store.
- Mac Provisioning Profile Signing Certificates: This type of Certificate may be used by Apple to sign provisioning profiles used for Mac application development and submission to the Mac App Store.
- Pass Certificates: This type of Certificate may be used by Developers authorized by Apple to digitally sign passes for Apple Wallet, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.
- Website Push Notification Certificates: This type of Certificate may be used by Developers authorized to digitally sign notification packages for websites on OS X, enabling them to be authenticated to an Apple device and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.
- OS X Server Authentication Certificates: This type of Certificate may be used by Developers to access WWDR APIs' through Developer Tools.
- VoIP Services Push Certificates: This type of Certificate may be used by Developers of VoIP apps to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates for VoIP apps.
- Apple Pay Merchant Certificates: This type of Certificate may be used by Apple to encrypt transaction data sent from Apple to a merchant developer.
- Apple Pay Pass Certificates: This type of Certificate may be used by Apple to digitally sign Apple Pay Apple Wallet passes, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.
- TestFlight Distribution Certificates: This type of Certificate may be used by Apple to sign TestFlight apps and third-party macOS software packages, enabling them to run on an Apple device.

- WatchKit Services Certificates: This type of Certificate may be used by Developers authorized by Apple to create and maintain SSL connectivity to the Apple Push Notification service, enabling them to push updates to watch complications.
- Apple Pay Provisioning Encryption Certificates: The type of Certificate may be used by Apple Pay participating issuers to encrypt data during the provisioning process.
- Enhanced Pass Certificates: This type of Certificate may be used by Developers authorized by Apple to digitally sign enhanced Apple Wallet passes containing an NFC dictionary, enabling them to be installed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.
- tvOS Application Signing Certificates: This type of Certificate may be used by Apple to sign a tvOS application bundle, enabling it to be submitted to the App Store on Apple TV.
- WWDR Apple Push Services Client Authentication G2 Certificates: This type of Certificate may be used by Developers authorized by Apple to create and maintain SSL connectivity to the Apple Push Notification service Development and Production environments.
- Apple Pay Client Authentication Certificates:  This type of Certificate may be used by Apple Pay Merchant Developers to provide SSL connectivity and authentication to the Apple Pay environments. Reliance on this type of Certificate by any party other than Apple is strictly prohibited.
- Apple Development Certificates*: This type of Certificate may be used by Developers authorized by Apple to digitally sign a software application enabling the application to be tested on an Apple Mac, iPhone, iPad, iPod touch, Apple Watch and/or AppleTV.
- Apple Distribution Certificates*: This type of Certificate may be used by a Developer to digitally sign software applications and macOS application bundles for submission to Apple, or for distribution of software applications to their internal employees.
- Swift Package Collection Certificate : This type of Certificate may be used by Developers authorized by Apple to sign Swift Package Collections.

- Order Type ID Signing Certificates : This type of Certificate may be used by Developers authorized by Apple to digitally sign Order payloads, enabling them to be viewed on an Apple device, and to create and maintain SSL connectivity to the Apple Push Notification service for notification of updates.
- Identity Processing Certificates : This type of Certificate may be used by Developers authorized by Apple to decrypt identity elements, enabling the data to be utilized to support functionality in applications.

Certificates used exclusively for functions internal to Apple products and/or Apple processes, such as device profile signing Certificates, event log signing Certificates, etc. are not included within the scope of this CPS.

### 2.10.2.Limitations on Use

The WWDR Sub-CAs will not allow its Certificates to be used to create a certification authority or to allow its private key to sign a Certificate issued by another certification authority.

Except for internal-use Certificates, any Certificates issued from the WWDR Sub-CAs shall not be used for any purpose that is not identified in this CPS § 2.10.1 as a permitted use.

## *2.11. OBLIGATIONS*

This section sets forth policies related to the obligations of the WWDR Sub-CAs.

### 2.11.1. General Sub-CA Obligations

The WWDR Sub-CAs shall:

- Conform its operations to the Apple CP and to this CPS as the same may be amended from time to time.
- Issue and publish Certificates in accordance with the Apple CP and this CPS.
- Revoke Certificates issued by the WWDR Sub-CAs, upon receipt of a valid request to revoke the Certificate from a person authorized to request such a revocation. The validity of the request and the authorization of the person making the request will be determined by the WWDR Sub-CAs.
- Publish Certificate Revocation Lists (CRLs) on a regular basis, or provide OCSP responses in accordance with the Apple CP. As applicable, the CA shall notify the Subscriber that the certificate has been revoked.

### 2.11.2. Notification of Issuance to Subscribers

For certificates issued to Developers, notification to Subscribers is deemed to have taken place when newly issued Certificates are made available via the Apple Developer website, Xcode or downloaded to the Subscriber's machine.

### 2.11.3. Notification of Issuance to Others

For certificates issued to Developers, notification to a Developer's Agent and/or Administrator is deemed to have taken place when newly issued certificates are made available via the Apple Developer website.

### 2.11.4. Notification of Revocation to Subscribers

An active Subscriber who requests a Certificate revocation is notified upon the revocation of the Certificate via the contact information associated with the Certificate.

### 2.11.5. Notification of Revocation to Others

Notification of revocation to others is deemed to have taken place upon publication of the CRL or update of the certificate status information in the OCSP Response.

### 2.11.6. Registration Authority ("RA") Obligations

An external RA is not used. For Certificates issued to Developers, the WWDR Sub-CAs perform limited RA services to provide reasonable assurance that Certificates are only issued to members of the Developer Program.

### 2.11.7. Subscriber Obligations

Subscribers are obligated to:

- Safeguard Developer account and their private key from compromise.
- Use their Certificates exclusively for legal purposes.
- Promptly request that a Certificate be revoked if the Subscriber has reason to believe there has been a compromise of the Certificate's associated private key.
- For Pass Certificates, Enhanced Pass Certificates, and Apple Pay Pass Certificates, a request for revocation is initiated by sending an email to product-security@apple.com.
- For OS X Server Authentication Certificates, revocation is initiated by disassociating the OS X Server from the team via Xcode, or via the Apple Developer website.

- For all other WWDR Sub-CA certificates issued to Developers, revocation is performed via the Apple Developer website. After authenticating to the website, the Subscriber follows the link to the revocation page and identifies the Certificate to be revoked. Only Certificates issued to the authenticated Subscriber, can be revoked based upon a request from such entity.
- Promptly request that a Certificate be revoked if the Subscriber is not authorized to use the applicable Certificate on behalf of the Subscriber entity (e.g. no longer employed by the Subscriber entity).
- Take no action to transfer their Certificate to any third party unless otherwise authorized by Apple.

### 2.11.8. Relying Party Obligations

Relying Parties are obligated to:

- Acknowledge that they are solely responsible for deciding whether or not to rely on the information in a Certificate, and agree that they have sufficient information to make an informed decision. Apple shall not be responsible for assessing the appropriateness of the use of a Certificate.
- Acknowledge that, to the extent permitted by applicable law, Apple hereby disclaims all warranties regarding the use of any Certificates, including any warranty of merchantability or fitness for a particular purpose. In addition, Apple hereby limits its liability and excludes all liability for indirect, special, incidental, and consequential damages.
- Restrict reliance on Certificates issued by the CA to the purposes for which those Certificates were issued, in accordance with the CP and this CPS.

# 3. KEY LIFE CYCLE MANAGEMENT

This section sets forth practices related to the key life cycle management controls of the WWDR Sub-CAs.

## *3.1. SUB-CA KEY PAIR GENERATION*

Sub-CA signing key generations occur using a secure cryptographic device meeting the requirements as described in CP §3.2. The signing key pair for the WWDR Sub-CA and the Developer Authentication Sub-CA is 2048-bits using the RSA algorithm. The signing key pair for the WWDR-G2 Sub-CA is a 256-bit Elliptic Curve Cryptography key.

Sub-CA signing keys are used to sign Certificates and Sub-CA CRLs.

The WWDR Sub-CAs private keys will cease to be used, and be replaced at the end of a designated period, up to a maximum of fifteen (15) years, or when a compromise is known or suspected.

## *3.2. SUB-CA PRIVATE KEY PROTECTION*

### 3.2.1. Sub-CA Private Key Storage

Sub-CA private keys are stored in a Hardware Security Module (HSM) that is tamper resistant and certified at a minimum level of FIPS 140-2 Level 3.

### 3.2.2. Sub-CA Private Key Control

There is a separation of physical and logical access to each Sub-CA private key, and a minimum of two individuals is required for physical access to the HSM where the Sub-CA private keys are stored.

### 3.2.3. Sub-CA Key Escrow

The WWDR Sub-CAs private keys shall not be placed in escrow.

### 3.2.4. Sub-CA Key Backup

The WWDR Sub-CAs private keys are backed up for recovery purposes. Backups are stored in a secured environment, and a minimum of two individuals are required for logical recovery.

### 3.2.5. Sub-CA Key Archival

The WWDR Sub-CAs shall archive any necessary keys for a period of time sufficient to support the responsibilities of the WWDR Sub-CAs.

## *3.3. SUB-CA PUBLIC KEY DISTRIBUTION*

Each Sub-CA public key will be contained in an X.509 Certificate signed by an Apple Root CA, that may be provided to Subscribers as necessary to support the WWDR PKI.

## *3.4. SUB-CA KEY CHANGEOVER*

When a new private key is required, a new Sub-CA signing key pair will be generated and all subsequently issued certificates and CRLs are signed with the new private signing key. The corresponding new Sub-CA public key Certificate may be provided to Subscribers as necessary to support the WWDR PKI.

## *3.5. SUBSCRIBER KEY PAIR GENERATION*

When a Subscriber is signed into Xcode with their Apple ID and Password and initiates an eligible request for a new Cloud Managed certificate, a new signing key pair will be generated in an Apple managed Hardware Security Module(HSM). The corresponding public key Certificate may be provided to Subscribers via Xcode or the developer website. The Sub-CA does not provide Subscriber key pair generation for any other WWDR certificates.

## *3.6. SUBSCRIBER PRIVATE KEY PROTECTION*

### 3.6.1. Subscriber Private Key Storage

Subscriber Cloud Managed private keys are generated in an Apple Managed Hardware Security Module (HSM) that is tamper resistant and certified at a minimum level of FIPS 140-2 Level 2. After generation, the Subscriber private key is encrypted with an additional key in the HSM and the Subscriber encrypted private key is stored in a separate database.

### 3.6.2. Subscriber Private Key Control

Subscriber Cloud Managed private keys are encrypted prior to storage. When WWDR receives eligible signing requests from a Subscriber, the Cloud managed encrypted private key is sent to an Apple managed HSM for decryption and processing signing requests. The decrypted private key does not leave the Apple managed HSM that is tamper resistant and certified at a minimum level of FIPS 140-2 Level 2.

# 4. CERTIFICATE LIFE CYCLE MANAGEMENT

This section sets forth practices related to the certificate life cycle management controls of the WWDR Sub-CAs.

## *4.1. EXTERNAL RA REQUIREMENTS*

An external Registration Authority is not utilized by the WWDR Sub-CAs.

## *4.2. CERTIFICATE REGISTRATION*

For Certificates that are not Cloud Managed, eligible Subscribers create a Certificate Signing Request ("CSR") using a corresponding private/public key pair generated on the client computer. For OS X Server Authentication, iOS Development Certificates, and Apple Development Certificates, the CSR is automatically generated and sent via Xcode when the Subscriber logs in with their Apple ID and password. For all other WWDR certificates, Subscribers upload the completed CSRs manually after logging into the Apple Developer website with their Apple ID and password.

For Certificates that are Cloud Managed, eligible Subscribers request a CSR to be generated in an Apple managed High Security Module (HSM) via Xcode. For iOS Development Certificates, and Apple Development Certificates, the CSR is automatically requested via Xcode when the Subscriber logs in with their Apple ID and password. For all other eligible Cloud Managed WWDR certificates, the CSR is automatically requested via Xcode when initiated by a Subscriber.

Upon receipt, the CSR is processed by the Apple Developer website. Except for iOS Development Certificates and Apple Development Certificates, the issuance of a Developer Certificate is contingent upon the requesting Subscriber being an eligible member of the Developer Program. The Apple Developer website verifies that the account is one that is eligible for the issuance of Apple Worldwide Developer Relations certificates and that, if applicable, the Developer Program subscription payments are current. iOS Development Certificates and Apple Development Certificates are issued to individuals with a valid Apple ID.

The name associated with a Certificate is either the individual Subscriber's name and organization, or the individual Subscriber's email address, or the Subscriber's phone number, or the organization name, or application identifier, or team identifier, only as applicable.

Certificates with Apple as the intended Subscriber are issued at the request of an authorized Apple employee in accordance with internal business practices.

## *4.3. CERTIFICATE RENEWAL*

For OS X Server Authentication certificates, Xcode will automatically initiate a new Certificate request when the existing certificate is close to expiring. A new certificate is generated if the Subscriber continues to be an eligible member of the Developer Program.

For Cloud Managed certificates, Xcode will automatically initiate a new Certificate request when an eligible request is received from the Subscriber and the existing certificate is close to expiring or has already expired. A new certificate is generated if the Subscriber continues to be an eligible member of the Developer Program.

For all other WWDR certificates, a new certificate can be issued at the request of the Subscriber using the same process used at initial Certificate issuance.

## *4.4. CERTIFICATE REKEY*

The WWDR Sub-CAs does not rekey certificates. Compromised keys result in completely new key sets and certificates being issued.

## *4.5. CERTIFICATE ISSUANCE*

Certificates are issued to the ISO 9594/X.509 standard and signed using one of the WWDR Sub-CAs signing keys.

## *4.6. CERTIFICATE ACCEPTANCE*

For Certificates issued to Developers, Certificates shall be deemed accepted and valid immediately upon issuance.

## *4.7. CERTIFICATE DISTRIBUTION*

For Certificate issued to Developers, Certificates will be distributed to the Developer through the Apple Developer website, or automatically downloaded to the Developer's machine via Xcode.

## *4.8. CERTIFICATE REVOCATION*

Certificates with Apple as the intended Subscriber are revoked at the request of an authorized Apple employee in accordance with internal business practices.

For Pass Certificates, Enhanced Pass Certificates, and Apple Pay Pass Certificates, the Subscriber may initiate a revocation request by sending an email to product-security@apple.com. The request for revocation will then be evaluated by Apple.

For all other certificates issued to Developers, except for iOS Submission Certificates issued to a Developer in the Apple Developer Enterprise Program, the certificate revocation process will commence upon receipt of a valid request to revoke the Certificate from the Subscriber via the Apple Developer website or via Xcode. iOS Submission Certificates can be revoked by a Developer in the Apple Developer Enterprise Program by submitting a valid revocation request via the Apple Developer website. The Subscriber will be required to log in using their Apple ID and password. After authentication, the Subscriber will indicate that they wish to revoke their Certificate.

Certificates may be revoked by Apple for any reason without notice to the Subscriber.

Revoked certificates are noted in one of the WWDR Sub-CAs CRLs, and/or via the Online Certificate Status Protocol.

## *4.9. CERTIFICATE SUSPENSION*

Certificate suspension (temporary revocation) is supported for Mac App Development Certificates, Apple Development Certificates, and iOS Submission Certificates. Apple can suspend a Certificate that is supported for suspension at Apple's discretion. Once suspended, the Certificate is revoked with a reason code certificateHold. At a minimum, a Certificate may remain suspended until the certificate expiration date. A Certificate may be unsuspended at Apple's discretion as part of a re-instatement request process via the Apple Developer website.

## *4.10. CERTIFICATE STATUS*

The WWDR Sub-CAs utilize two methods for certificate validation, Certificate Revocation List (CRL) and Online Certificate Status Protocol (OCSP). Refer to the CRL Distribution Point ("CDP") or the Authority Information Access ("AIA") extension in the Certificates for the status information method used.

### 4.10.1. CRL Usage

Subscribers may use a CRL, which is updated periodically at Apple's sole discretion, to determine the status of a particular Certificate containing a CRL Distribution Point (CDP) extension. At a minimum, revoked Certificates remain in the CRL until the Certificates have expired with the exception of when a Certificate is unsuspended. More than one valid CRL Certificate may exist at one time.

Certificates and CRLs issued by the WWDR Sub-CAs shall be retained for a period of not less than two (2) years.

### 4.10.2.OCSP Usage

Subscribers may use OCSP to determine the status of a particular Certificate containing an Authority Information Access (AIA) extension. At a minimum, revoked Certificates remain marked as "revoked" for the certificate lifetime with the exception of when a Certificate is unsuspended. A delegate leaf Certificate is used to sign all OCSP responses. This leaf is signed by one of the WWDR Sub-CAs private keys.

OSCP status requests must contain at a minimum the certificate serial number to receive a valid response. Once an OCSP request has been validated there will be a signed response back to the requestor indicating the status of the Certificate and showing the request was successful. Failed OCSP requests will generate a failure status back to the requestor.

## *4.11. CERTIFICATE PROFILE*

Certificates issued by the WWDR Sub-CA shall conform to the X.509 version 3 Certificate format, and shall contain the following elements:

| Field/Attribute | Value |
|---|---|
| Issuer DN | C = US, O = Apple Inc., OU = Apple Worldwide Developer Relations, CN = Apple Worldwide Developer Relations Certification Authority |
| | CN = Apple Worldwide Developer Relations CA - G2, OU= Apple Certification Authority, O = Apple Inc., C = US |
| | CN = Apple Worldwide Developer Relations Certification Authority, OU = G3, O = Apple Inc., C = US |
| | CN = Apple Worldwide Developer Relations Certification Authority, OU = G4, O = Apple Inc., C = US |
| | CN = Apple Worldwide Developer Relations Certification Authority, OU = G5, O = Apple Inc., C = US |
| | CN = Apple Worldwide Developer Relations Certification Authority, OU = G6, O = Apple Inc., C = US |

| | |
|---|---|
| CRL Distribution Points and/or Certificate Authority Information Access | URL of the location where a Relying Party can check the status of a certificate. |

Individual WWDR Sub-CA certificate profiles also contain the following:

### 4.11.1. iOS Development and Submission Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | iPhone Software Submission Signing (1.2.840.113635.100.6.1.4) |
| | Yes | iPhone Software Development Signing (1.2.840.113635.100.6.1.2) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.2. APNs SSL Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1 with RSA Encryption |
| Key Usage | No | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple Push Notification service Development (1.2.840.113635.100.6.3.1) |

| | | |
|---|---|---|
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| Basic Constraints | No | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.3. Push CSR Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Custom Extensions | No | CSR Signing (1.2.840.113635.100.4.12) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.4. Safari Extension Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Safari Extension Signing (1.2.840.113635.100.4.8) |
| Custom Extensions | Yes | Safari Extension Signing (1.2.840.113635.100.6.1.5) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.5.Mac Application Development Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | Mac Application Software Development Signing (1.2.840.113635.100.6.1.12) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.6.Mac Application Submission Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | Mac Application Software Submission Signing (1.2.840.113635.100.6.1.7) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.7.Mac Installer Package Submission Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Mac Installer Package Signing (1.2.840.113635.100.4.9) |
| Custom Extensions | Yes | Mac Installer Package Submission Signing (1.2.840.113635.100.6.1.8) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.8.Mac App Store Application Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | No | Mac App Store Application Software Signing (1.2.840.113635.100.6.1.9) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.9.Mac App Store Installer Package Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2with RSA Encryption |
| Key Usage | Yes | Digital Signature |

| | | |
|---|---|---|
| Extended Key Usage | Yes | Mac App Store Installer Package Signing (1.2.840.113635.100.4.10) |
| Custom Extensions | No | Mac Installer Package Submission Signing (1.2.840.113635.100.6.1.10) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.10.Mac App Store Receipt Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1 with RSA/ SHA-2 with ECDSA Encryption |
| Key Usage | Yes | Digital Signature |
| Custom Extensions | No | Mac App Store Receipt Signing (1.2.840.113635.100.6.11.1) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Mac App Store Receipt Signing Policy (1.2.840.113635.100.5.6.1) |

### 4.11.11.Mac Provisioning Profile Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1/SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Custom Extensions | No | Mac OS X Provisioning Profile Signing (1.2.840.113635.4.11) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.12.Pass Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-1 with RSA Encryption |
| Key Usage | No | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.4.14) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.1.16) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| Basic Constraints | No | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.13.Website Push Notification Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.4.15) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.1.17) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.14. VoIP Services Push Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple Push Notification service Development (1.2.840.113635.100.6.3.1) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.6.3.3) |
| | No | Apple custom extension (1.2.840.113635.100.6.3.4) |
| | No | Apple custom extension (1.2.840.113635.100.6.3.5) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.15. Apple Pay Pass Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.4.14) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.1.16) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |

| | | |
|---|---|---|
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.16.TestFlight Distribution Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| | Yes | Apple Code Signing Third Party (1.2.840.113635.100.4.1.3) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.1.25.1) |
| | No | Apple custom extension (1.2.840.113635.100.6.1.25.2) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.17.WatchKit Services Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple Push Notification service Development (1.2.840.113635.100.6.3.1) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.6.3.5) |

| | | |
|---|---|---|
| | No | Apple custom extension (1.2.840.113635.100.6.3.6) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.18.Enhanced Pass Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | No | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.4.14) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.1.16) |
| | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |
| | No | Apple custom extension (1.2.840.113635.100.6.1.26) |
| Basic Constraints | No | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.19.WWDR Apple Push Services Client Authentication G2 Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | No | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2) |

| | | |
|---|---|---|
| | No | Apple Push Notification service Development (1.2.840.113635.100.6.3.1) |
| | No | Apple custom extension (1.2.840.113635.100.6.3.6) |
| Basic Constraints | No | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.20.Apple Pay Merchant Client Authentication Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.32) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.21.Apple Pay Merchant Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | ECDSA-SHA256 |
| Key Usage | Yes | Key encipherment<br>Key agreement |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.32) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.22.Apple Pay Provisioning Encryption Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | ECDSA-SHA256 |
| Key Usage | Yes | Key encipherment<br>Key agreement |
| Custom Extensions | Yes | Apple custom extension (1.2.840.113635.100.6.39) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.23.tvOS Application Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | ECDSA-SHA256 |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | Apple custom extension (1.2.840.113635.100.6.1.24) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.24.Apple Development Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | iPhone Software Development Signing<br>(1.2.840.113635.100.6.1.2) |

| | | |
|---|---|---|
| | Yes | Mac Application Software Development Signing (1.2.840.113635.100.6.1.12) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.25.Apple Distribution Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | Yes | iPhone Software Submission Signing (1.2.840.113635.100.6.1.4) |
| | Yes | Mac Application Software Submission Signing (1.2.840.113635.100.6.1.7) |
| | No | Cloud Managed (1.2.840.113635.100.6.1.32) Optional |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.26.Swift Package Collection Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | Yes | Code Signing (1.3.6.1.5.5.7.3.3) |
| Custom Extensions | No | Swift Package Collection Signing (1.2.840.113635.100.6.1.35) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.27.Order Type ID Signing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication<br>Apple custom Extended Key Usage (1.2.840.113635.100.4.19) |
| Custom Extensions | No | Apple Push Notification service Production (1.2.840.113635.100.6.3.2)<br>Apple Order Payload Signing (1.2.840.113635.100.14.2) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

### 4.11.28.Identity Processing Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with ECDSA Encryption |
| Key Usage | Yes | Key Agreement |
| Custom Extensions | No | Apple custom extension<br>(1.2.840.113635.100.6.75) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy<br>(1.2.840.113635.100.5.1) |

Certificates issued by the Developer Authentication Sub-CA shall conform to the X.509 version 3 Certificate format, and shall contain the following elements:

| Field/Attribute | Value |
|---|---|
| Issuer DN | C = US, O = Apple Inc., OU = Apple Worldwide Developer Relations, CN = Developer Authentication Certification Authority |
| CRL Distribution Points and/or Certificate Authority Information Access | URL of the location where a Relying Party can check the status of a certificate. |

Individual Developer Authentication Sub-CA certificate profiles also contain the following:

### 4.11.29.OS X Server Authentication Certificates

| Field/Attribute | Critical | Value |
|---|---|---|
| Signature Algorithm | N/A | SHA-2 with RSA Encryption |
| Key Usage | Yes | Digital Signature |
| Extended Key Usage | No | Client Authentication (1.3.6.1.5.5.7.3.2) |
| Custom Extensions | No | Apple custom extension (1.2.840.113635.100.6.21) |
| Basic Constraints | Yes | Certification Authority = No |
| Certificate Policy | No | Apple Certificate Policy (1.2.840.113635.100.5.1) |

## *4.12. CRL PROFILE*

A CRL issued by one of the WWDR Sub-CAs shall conform to the X.509 version 2 CRL format. Each CRL shall contain the following fields

- Signature Algorithm of SHA1/SHA-2 with RSA Encryption, or SHA-2 with ECDSA
- Issuer matching the Sub-CA Certificate's Distinguished Name
- This Update with the time of CRL issuance
- Next Update defining the period of validity
- Authority Key Identifier extension
- List of Revoked Certificates

## *4.13. INTEGRATED CIRCUIT CARDS*

Not applicable.

# 5. ENVIRONMENTAL CONTROLS

This section sets forth practices related to the environmental controls of the WWDR Sub-CAs.

## *5.1. CPS ADMINISTRATION*

Apple has designated a management group called the Policy Authority (PA) with final authority and responsibility for specifying and approving this CPS.

This authorized body has performed an assessment to evaluate business risks and determine the security requirements and operational procedures to be included in the CPS for the following:

- Key life cycle management controls
- Certificate life cycle management controls
- CA environmental controls

The CPS is made publicly available to all Subscribers and Relying Parties, including any revisions that occur from time to time.

Any changes to the CPS, along with the effective date of the changes, shall be reviewed by the PA, and posted in a timely manner.

## *5.2. CA TERMINATION*

As set forth in this section, any decision to terminate any one of the WWDR Sub-CAs shall be approved by the Policy Authority and an Apple Vice President prior to the effective date of termination.

At the time of termination, Apple will develop a termination plan addressing the following:

- Provision of notice to related parties affected by the termination,
- Revocation of certificates issued by the Sub-CA,
- Preservation of the Sub-CA's archives and records

## *5.3. CONFIDENTIALITY*

Apple shall keep the following information confidential at all times:

- All private signing and client authentication keys
- Security and annual audits and security parameters

- Personal or non-public information about Subscribers
- Security mechanisms

Except as required to support the WebTrust audit performed by an independent external audit firm, confidential information should not be released to third parties unless required by law or requested by a court with jurisdiction over the CA. The information will be kept confidential even after the termination of the CA.

- The following information shall not be considered confidential:
- Information included in Certificates
- Any of the WWDR Sub-CAs public Certificates
- Information contained in the CPS and CP documents
- Any Certificate status or Certificate revocation reason code

## *5.4. INTELLECTUAL PROPERTY RIGHTS*

Certificates and CRLs issued by the WWDR Sub-CAs, information provided via OCSP, the CPS, and the CP are the property of Apple.

## *5.5. PHYSICAL SECURITY*

Physical protection of equipment supporting the WWDR PKI is achieved through the creation of clearly defined security perimeters with appropriate physical barriers to entry. Details of the physical security policies and procedures are in appropriate internal security documents.

Equipment is located or protected to reduce the risks from environmental threats and hazards, including but not limited to power and air conditioning disruption or failure, water exposure, fire, telecommunications disruption or failure and opportunities for unauthorized access.

At end of life, cryptographic devices are physically destroyed or zeroized in accordance to manufacturers' guidance prior to disposal.

## *5.6. BUSINESS CONTINUITY MANAGEMENT*

Business continuity plans have been developed to maintain or restore the Sub-CA business operations in a timely manner following interruption or failure of critical business processes.

## *5.7. EVENT LOGGING*

### 5.7.1. Archiving

Event journal data is archived on a periodic basis.

A risk assessment has been performed to determine the appropriate length of time for retention of archived event journals.

Archived event journals are maintained at a secure location for a predetermined period.

### 5.7.2. Event Journal Reviews

Current or archived event journals may only be retrieved by authorized individuals and only for valid business or security reasons.

Event journals are reviewed periodically.

The review of current and archived event journals includes the identification and follow-up of exceptional, unauthorized, or suspicious activity.

# 6. REVISION HISTORY

| Issue Number | Issue Date | Details |
|---|---|---|
| 1.0 | 03/06/08 | Initial release. |
| 1.1 | 03/17/09 | Updates to reflect the addition of Apple Push Notification service Client SSL Certificates. |
| 1.2 | 06/07/10 | Updates to reflect the addition of iPad, the Authority Information Access extension to Developer Identification Certificates and Safari Extension Signing Certificates. |
| 1.3 | 10/20/10 | Updates to reflect the addition of Mac Application and Mac Installer Package Submission Certificates. |
| 1.4 | 01/06/11 | Updates to reflect the addition of Mac App Store Application, Mac App Store Installer Package, and Mac App Store Receipt Signing Certificates. |
| 1.5 | 07/18/11 | Updates to reflect the addition of the Mac Provisioning Profile Signing Certificate. |
| 1.6 | 09/02/11 | Updates to reflect the addition of the Mac Application Development Certificate. |
| 1.7 | 10/04/11 | Updates to reflect the addition of the Push CSR Signing Certificate. |
| 1.8 | 06/11/12 | Updates to reflect the addition of the Pass Certificate.<br>Minor updates to clarify the roles of existing certificate types. |
| 1.9 | 03/21/13 | Updates to reflect changes to the revocation process for Pass Certificates.<br>Updated format of certificate profile descriptions. |
| 1.10 | 06/10/13 | Updates to reflect the addition of the Website Push Notification Certificates and the addition of the Developer Authentication Sub-CA and OS X Server Authentication Certificate Profile.<br>Clarifications to the CA termination process. |

| | | |
|---|---|---|
| 1.11 | 06/03/14 | Updates to reflect the addition of the VoIP Services Push Certificates. |
| 1.12 | 10/14/14 | Updates to reflect the addition of the Apple Pay Merchant Certificates and Apple Pay Passbook Certificates and the Apple Worldwide Developer Relations CA – G2 Sub-CA. |
| 1.13 | 06/08/15 | Updates to reflect the addition of a SHA-2 version of the iOS Development Certificate, and the addition of the new TestFlight Distribution Certificate, the WatchKit Services Certificate, and the Apple Pay Provisioning Encryption Certificate. |
| 1.14 | 09/09/15 | Updates to reflect the addition of the Enhanced Pass Certificates and updates to the Developer Program. |
| 1.15 | 04/01/16 | Updates to reflect the addition of the SHA-2 versions of the Mac OS Provisioning Profile Signing, Mac OS Application, and Mac OS Installer Package Certificates. Updates to reflect the addition of the tvOS Application Signing Certificate and the WWDR Apple Push Services Client SSL G2 Certificates. Updates to reflect renaming of Passbook to Apple Wallet. Updates to reflect changes to the issuance of iOS Development Certificates. |
| 1.16 | 06/13/2016 | Updates to reflect the addition of the Apple Pay Merchant Client Authentication Certificates. Renaming of the WWDR Apple Push Services Client SSL G2 Certificates to WWDR Apple Push Services Client Authentication G2 Certificates. Updated 'Apple Pay Wallet' to Apple Wallet to reflect current name. |
| 1.17 | 03/20/2017 | Updates to reflect change from OS X to macOS. OS X Server references maintained. For certificate revocations, modified policy to allow for certificate revocations by Apple without notice. |

| | | |
|---|---|---|
| 1.18 | 08/16/2017 | Revised revocation notifications to reflect notifications via the contact information associated with the Subscriber's certificate.<br>Added phone number as a Subscriber identifier.<br>Changed font to SF Hello. |
| 1.19 | 03/20/2019 | Updates to reflect Contact Details and annual review. |
| 1.20 | 06/03/2019 | Updates to reflect the addition of the Apple Development Certificates and Apple Distribution Certificates. |
| 1.21 | 02/19/2020 | Updates to reflect the addition of the new WWDR Sub-CA. |
| 1.22 | 11/04/2020 | Updated section Certificate revocation and Certificate status.<br>Added section Certificate suspension. |
| 1.23 | 01/13/2021 | Updated section Certificate Profile to reflect changes in signature algorithm. |
| 1.24 | 06/02/2021 | Added two new WWDR Sub-CAs. Added support for ECDSA for App Store Receipt Signing Certificates. Updated TestFlight Distribution Certificate with a new optional custom EKU. Updated various sections to reflect changes related to Cloud Managed Certificates. |
| 1.25 | 08/10/2021 | Updates to reflect the addition of the Swift Package Collection certificate. |
| 1.26 | 01/14/2022 | Added new WWDR Sub-CA in section 1 and 4.11. Updated CRL profile to reflect change in signature algorithm in section 4.12. |
| 1.27 | 06/15/2022 | Added new Certificates Order Type Signing and Identity Processing. |