

# Table of Contents

## Apple Plattformen 27

**Stand: 09. Juni 2026 - Beta 1 aktiv (WWDC 2026 - Platforms State of the Union)** Apple hat iOS 27, iPadOS 27, macOS 27 (Golden Gate), watchOS 27 und weitere Plattformen während der WWDC-2026-Keynote am 8. Juni 2026 offiziell angekündigt. Developer Beta 1 steht unmittelbar nach der Keynote zum Download bereit. Alle Build-Nummern bestätigt via [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/) am 08. Juni 2026. Aktualisiert am 09. Juni 2026 mit Erkenntnissen aus dem WWDC Platforms State of the Union.

---

## Übersicht

Plattform

Stabil (vor WWDC)

Beta 1 Build

Beta 1 Datum

Release (erwartet)

iOS

26.5.1 (23F81)

**24A5355q** ✓

08. Juni 2026

September 2026

iPadOS

26.5.1 (23F81)

**24A5355q** ✓

08. Juni 2026

September 2026

macOS Golden Gate

26.5.1 (25F80)

**26A5353q** ✓

08. Juni 2026

Herbst 2026

watchOS

26.5 (23T570)

**24R5289n** ✓

08. Juni 2026

September 2026

tvOS

26.5

**24J5289o** ✓

08. Juni 2026

September 2026

visionOS

26.5

**24M5291p** ✓

08. Juni 2026

September 2026

Xcode

26.5 (17F42)

**27A5194q** ✓

08. Juni 2026

Herbst 2026

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/) – bestätigt am 08. Juni 2026. Swift 6.4 in Xcode 27 Beta 1 enthalten (bestätigt via [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)).

## Wichtige Termine:

Datum

Ereignis

08.-12. Juni 2026

WWDC 2026 (Apple Park, hybrid) – „All Systems Glow”

08. Juni 2026, ca. 13:00 Uhr PT

Developer Beta 1 (alle Plattformen)  Bestätigt

Juli 2026

Public Beta (erwartet)

September 2026

Finale Release (erwartet)

April 2027

App Store: iOS 27 SDK Pflicht (erwartet)

**Fokus dieser Dokumentation:** Delta zu Apple Plattformen 26 – was ist neu, was ändert sich, wo müssen Entwickler aktiv handeln.


**WWDC-Motto:** „All Systems Glow”

---

## iOS 27

### Beta 1

**Datum:** 08. Juni 2026  Bestätigt

**Build-Nummer:** 24A5355q  Bestätigt

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), WWDC 2026 Keynote, MacRumors, 9to5Mac, TechTimes

**Siri-Neugestaltung**  Bestätigt

Apple hat Siri grundlegend neu gestaltet. Die neue „Siri AI” (Chatbot-Modus) nutzt **Google Gemini** als primäres Backend-Modell für komplexe Konversationsanfragen. ⚠️ Klarstellung: Apples eigene On-Device KI (Apple Intelligence / Foundation Models) bleibt unverändert lokal. Gemini ist das Modell hinter dem neuen Siri-Chatbot-Interface, nicht hinter Apple Intelligence.

**Neue Siri-Funktionen:** - **Eigenständige Siri-App** mit Chatbot-Interface (iMessage-ähnliche Konversationsdarstellung) - **Dynamic Island Integration:** Siri-Animation erscheint in der Dynamic Island (statt am unteren Bildschirmrand) - **Neue Aktivierungsmethode:** Vom Homescreen nach unten wischen öffnet das „Search or Ask“-Interface - **On-screen Awareness:** Siri erkennt und interagiert mit aktuell sichtbaren Bildschirminhalten - **Personal Context:** Siri kennt Kontakte, Kalender, E-Mails, Fotos und Dateien ohne wiederholte Eingabe - **Spotlight-Integration:** Suche und KI-Abfragen über Spotlight zusammengeführt - **Stimmenpassung:** Sprechtempo und Ausdrucksstärke einstellbar - **Sprachunterstützung initial:** Englisch, weitere Sprachen folgen - **Cross-Platform:** Verfügbar auf iOS, iPadOS, macOS, watchOS, visionOS, CarPlay und AirPods

**Relevanz für Entwickler:** - App Intents sind jetzt die primäre und einzige empfohlene Integrationsfläche für Siri - SiriKit erhält eine formale Deprecation-Notice (Migration innerhalb 2-3 Jahre) - Apps ohne App-Intents-Integration können für die neue Gemini-gestützte Siri unsichtbar sein

### *Siri Extensions API* **Bestätigt**

iOS 27 führt die **Siri Extensions API** ein. Drittanbieter-KI-Dienste (Google Gemini, Anthropic Claude, OpenAI ChatGPT) können direkt in Siri, Writing Tools und Image Playground eingebunden werden. Nutzer wählen ihren bevorzugten KI-Dienst in den Systemeinstellungen.

**Relevanz für Entwickler:** - Neue SiriExtension-Schnittstelle implementieren, um den eigenen KI-Dienst als Siri-Alternative anzubieten - Ein dedizierter App-Store-Bereich für Extensions-kompatible Apps ist geplant

### *Core AI Framework* **Bestätigt**

Apple ersetzt **Core ML** durch ein neues **Core AI Framework**. Das neue Framework bietet: - Standardisierter Plugin-Mechanismus für externe KI-Modelle - Model Context Protocol (MCP) als Connector-Schnittstelle - Integration beliebiger MCP-kompatibler Modelle in Apps

**Relevanz für Entwickler:** - Core ML wird deprecated – mittelfristige Migration auf Core AI Framework notwendig - Bestehende Core-ML-Implementierungen funktionieren weiter (Übergangszeit) - Neue KI-Funktionen sollen über Core AI Framework implementiert werden

### *Design: Liquid Glass Refinements* **Bestätigt**

**System-weiter Opacity-Slider:** Nutzer können Intensität des Liquid-Glass-Effekts systemweit anpassen

Adressiert Accessibility-Kritik: Kontrast und Lesbarkeit für Nutzer mit Sehbeeinträchtigungen verbessert

**Tab-Bar-Suche re-integriert:** In iOS 26 ausgelagerter Such-Button kehrt in die Tab Bar zurück

Betrifft Music, TV, Podcasts, News, Health auf Systemebene

**Relevanz für Entwickler:** - .glassEffect() Modifier und GlassEffectContainer erhalten Compositing-Pipeline-Updates - Diese propagieren automatisch auf Apps, die den Modifier bereits verwenden - Mixed UIKit/SwiftUI Apps: GlassEffectContainer-Grenzen manuell auf konsistentes Morphing prüfen - Apps mit UIGlassEffect via UIVisualEffectView: Manuelle Updates erforderlich - Custom Tab Bars und Floating-Search-Buttons aus iOS-26-Implementierungen sofort gegen iOS-27-Beta testen

### *Performance-Verbesserungen* **Bestätigt**

Apple fokussiert iOS 27 stark auf Stabilität und Performance – vergleichbar mit dem „Snow Leopard“-Ansatz (macOS X 10.6, 2009):

Bereich

Verbesserung

AirDrop-Transfers

bis zu 80% schneller

App-Starts

bis zu 30% schneller

Kamera-Fotoladen

bis zu 70% schneller

Mail-Laden

signifikant beschleunigt

Apple Music

beschleunigtes Laden

CPU-Scheduler

optimiert ab iPhone 11

**Neue Nutzer-Features**  **Bestätigt**

**Eigenständige Passwörter-App:** Kann Passwörter automatisch ändern

**Photos: AI-Bearbeitung** – Clean Up überarbeitet; neue Werkzeuge Extend, Reframe

**Visuelle Intelligenz:** Ernährungsscanner (Nährwerte scannen), Visitenkarten, Veranstaltungstickets

**Apple Wallet:** Benutzerdefinierte digitale Pässe aus physischen Karten erstellen

**Apple Cash:** KI-gestützte Rechnungsteilung via Fotobelege

**Generierte Untertitel:** Automatische Untertitel für persönliche iPhone-Videos

**Wallpaper-Generator:** KI-generierte Wallpaper via Apple Intelligence

**Shortcuts:** Natural Language Shortcut Creation

**Safari:** Tab-Automatisierung mit KI, „Search or Ask“ in Spotlight

**iCloud Shared Albums:** Volle Auflösung, funktioniert auf Android und Windows

**Gebärdensprach-Interpretation (FaceTime):** Neue API für Gebärdensprach-Erkennung

**iPhone Apps: Größenänderung auf iPad und iPhone Mirroring**  **Bestätigt**

**Quelle:** WWDC 2026 Platforms State of the Union (09. Juni 2026)

iPhone-Apps unterstützen ab iOS/iPadOS 27 echtes Resizing auf dem iPad und in iPhone Mirroring: - Apps können ihre Fenstergröße verändern (kein statisches Vollbild mehr) - SwiftUI-Layout-Engine passt Inhalte automatisch an - UIKit-Apps: Adaptive-Layout-Best-Practices kritisch

**Relevanz für Entwickler:** - Apps mit hartkodierter Fenstergröße oder fixen Frames sofort testen - GeometryReader / Size-Classes korrekt verwenden - UIResponsiveWindowSceneDelegate bei Bedarf implementieren

**Faltbares iPhone (Adaptive Layout APIs)**  **Bestätigt**

Apple hat **Adaptive Layout APIs** für faltbare Formfaktoren veröffentlicht, um Entwicklern einen Vorsprung für das erwartete iPhone Fold zu geben:

Hinge-State-Detection

Multi-Display-Konfiguration

Neue UIKit- und SwiftUI-Komponenten für adaptive Layouts

**Relevanz für Entwickler:** - UISplitViewController und Sidebar-Navigation jetzt implementieren - NavigationSplitView für faltbare Displays evaluieren - Adaptive Layout APIs testen, bevor Hardware (erwartet: Ende 2026 / Anfang 2027) verfügbar ist

**Child Safety**  **Bestätigt**

Überarbeitete Kindersicherungsfunktionen plattformübergreifend

Neue Screen Time-Erweiterungen

Neue Safari-Website-Request-Controls

**EU Regulatory: AirPlay-Alternative**  **Bestätigt**

Im Rahmen des EU Digital Markets Act: Google Cast als Alternative zu AirPlay integrierbar.

**Unterstützte Geräte**   
**Bestätigt**

**Korrektur (WWDC 2026 Keynote):** iOS 27 unterstützt iPhone 11 und neuer sowie iPhone SE (2. Gen.) und neuer — A13-Bionic-Geräte werden weiterhin unterstützt.

Gerät

Chip

iOS 27

iPhone 11

A13 Bionic

 Unterstützt

iPhone 11 Pro

A13 Bionic

 Unterstützt

iPhone 11 Pro Max

A13 Bionic

 Unterstützt

iPhone SE (2. Gen)

A13 Bionic

 Unterstützt

**Mindestvoraussetzung iOS 27:** A13 Bionic (iPhone 11 / iPhone SE 2. Gen und neuer)

## *SiriKit Deprecation* **Bestätigt**

Apple hat SiriKit auf WWDC 2026 formal deprecated. Entwickler haben ca. **2-3 Jahre** bis zum vollständigen Wegfall der Siri-Funktionalität für SiriKit-basierte Apps.

Deprecated

Nachfolger

SiriKit

App Intents

**App Intents (jetzt verpflichtend für neue Siri-Funktionen):** - Swift-natives Framework (seit iOS 16) - Metadaten werden zur Build-Zeit erzeugt und im App-Bundle gespeichert - OS versteht App-Fähigkeiten ohne App-Start - Streaming-Antworten (Multi-Turn-Dialog in einem App-Kontext) - Reichhaltigere Entity-Typen - Per-Intent Privacy Declarations (Cloud-Routing vs. On-Device)

### *Privacy: Neue APIs* **Bestätigt**

Neue Privacy-Manifest-APIs erlauben es Entwicklern zu deklarieren, ob Siri-Interaktionen on-device bleiben oder in die Cloud geroutet werden dürfen.

**Enterprise- und Healthcare-Apps:** Pflicht-Audit gegen neue Privacy-Manifest-APIs.

---

## Migration

## *UIScene-Lifecycle (Pflicht)* **Bestätigt**

**Quelle:** Apple Tech Note TN3187

**Risiko:** App startet ohne Adoption nicht mehr beim Build gegen iOS-27-SDK.

Deprecated AppDelegate

Ersatz SceneDelegate

applicationDidBecomeActive(\_:)

sceneDidBecomeActive(\_:)

applicationWillResignActive(\_:)

sceneWillResignActive(\_:)

applicationDidEnterBackground(\_:)

```

sceneDidEnterBackground(_:)
applicationWillEnterForeground(_:)
sceneWillEnterForeground(_:)
application(_:open:options:)
scene(_:openURLContexts:)
UIWindow.init(frame:)
UIWindow(windowScene:)

```

**Notwendige Schritte:** 1. SceneDelegate erstellen 2. UIApplicationSceneManifest in Info.plist eintragen 3. UIWindow-Initialisierung migrieren

*// Vorher (AppDelegate)*

```

func application(_ app: UIApplication,
    didFinishLaunchingWithOptions options: [UIApplication.LaunchOptionsKey: Any]?) ->
    Bool {
    window = UIWindow(frame: UIScreen.main.bounds)
    return true
}

```

*// Nachher (SceneDelegate)*

```

func scene(_ scene: UIScene, willConnectTo session: UISceneSession,
    options connectionOptions: UIScene.ConnectionOptions) {
    guard let windowScene = (scene as? UIWindowScene) else { return }
    window = UIWindow(windowScene: windowScene)
}

```

*Liquid Glass - Kompatibilitäts-Flag entfällt*  *Bestätigt*

**Quelle:** Apple Developer Documentation (iOS 26)

**Risiko:** UI-Rendering-Probleme in bestehenden Apps

Das Flag UIDesignRequiresCompatibility = true entfällt. Alle gegen iOS-27-SDK gebauten Apps erhalten automatisch Liquid Glass.

Komponente

Verhalten in iOS 27

UINavigationController

Transluzentes Glas-Material

UITabBar

Floating-Design, automatische Minimierung

UIToolbar

Automatische Gruppierung in Glas-Cluster

UISegmentedControl

Glas-Selektionseffekte

**Typische Probleme ohne Migration:** - Custom Backgrounds stören Glaseffekte - Automatisch eingefügte UIDropShadowView bricht Layouts - Floating Tab Bar verändert safeAreaInsets

**SceneKit vollständig deprecated**  **Bestätigt**

SceneKit ist vollständig deprecated. Migration zu RealityKit ist notwendig.

**UIScreen.main deprecated**  **Bestätigt**

*// Deprecated*

```
let size = UIScreen.main.bounds
```

*// Korrekt (via windowScene)*

```
let size = view.window?.windowScene?.screen.bounds
```

**SwiftUI actionSheet deprecated**  **Bestätigt**

*// Nicht mehr verwenden*

```
.actionSheet(isPresented: $show) { ... }
```

*// Korrekt*

```
.confirmationDialog("Titel", isPresented: $show) { ... }
```

**TLS 1.3 als Standard**  **Bestätigt**

TLS 1.3 ist Standard. Third-Party-SDKs auf Kompatibilität prüfen.

**SiriKit → App Intents**  **Bestätigt (Deprecation)**

Zeitfenster: ca. 2–3 Jahre. Migration von SiriKit auf App Intents jetzt einplanen.

**Core ML → Core AI Framework**  **Bestätigt (Deprecation)**


Core ML wird durch Core AI Framework abgelöst. Neue Features nur noch via Core AI Framework. Bestehende Implementierungen funktionieren weiterhin (Übergangszeitraum).

---

## iPadOS 27

### Beta 1

**Datum:** 08. Juni 2026  **Bestätigt**

**Build-Nummer:** 24A5355q  **Bestätigt** (identisch mit iOS 27 Beta 1)

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), WWDC 2026 Keynote

Alle iOS-27-Features gelten gleichermaßen für iPadOS 27.

**Adaptive Layout APIs für iPhone Fold**  **Bestätigt**

Die neuen Adaptive Layout APIs für faltbare Formfaktoren gelten auch für iPadOS 27. Sidebar-Navigation und Split-View-Erweiterungen profitieren von den gleichen neuen UIKit- und SwiftUI-Komponenten.

### **Performance-Verbesserungen** **Bestätigt**

Gleiche Performance-Verbesserungen wie iOS 27 (App-Starts, AirDrop, Kamera, Suche).

### **Siri / Apple Intelligence** **Bestätigt**

Alle iOS-27-Siri- und Apple-Intelligence-Features gelten für iPadOS 27.

---

## **Migration**

Die Migrationspunkte aus iOS 27 gelten vollständig für iPadOS 27.

---


## **macOS 27**

**Name: macOS Golden Gate**  **Bestätigt**

**Quelle:** WWDC 2026 Keynote (MacRumors, TechRadar, Shacknews)

### **Beta 1**

**Datum:** 08. Juni 2026  **Bestätigt**

**Build-Nummer:** 26A5353q  **Bestätigt**

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), WWDC 2026 Keynote

**Design: Liquid Glass Refinements**  **Bestätigt**

**Einheitliche Toolbar:** Apps erhalten eine einheitliche Toolbar am oberen Rand

**Vollbreite Sidebar:** Sidebar expandiert bis zur Fensterkante

**Schärferer Eckradius:** Straffere Corner Radius im Vergleich zu macOS 26

**Aktualisierte App-Icons:** Überarbeitetes Icon-Design

**Opacity-Slider:** System-weiter Regler für Liquid-Glass-Transparenz

Adressiert Feedback zu „Shadows and transparency quirks“ aus macOS 26 Tahoe

**Performance & Stabilität**  **Bestätigt**

macOS Golden Gate fokussiert stark auf Performance und Stabilität – Apple hat diesen Ansatz explizit mit macOS X Snow Leopard (2009) verglichen: - Schnellere User-Account-Erstellung - Dutzende optimierte Systemtechnologien

**Siri / Apple Intelligence**  **Bestätigt**

Alle iOS-27-Siri-Features auf macOS verfügbar (eigenständige Siri-App, Personal Context, On-Screen Awareness)

Spotlight: „Search or Ask“-Interface mit Siri AI

Apple Intelligence: Personal Context und On-Screen Awareness

**Photos**  **Bestätigt**

Überarbeiteter Clean Up sowie neue KI-Werkzeuge Extend und Reframe auch auf macOS.

**Safari**  **Bestätigt**

Tab-Gruppen mit KI-Automatisierung; „Search or Ask“ in Safari-Suche integriert.

### **Child Safety** **Bestätigt**

Überarbeitete Kindersicherungsfunktionen plattformübergreifend auch auf macOS.

### **MacBook Pro Touchscreen-Unterstützung (Nicht von Apple bestätigt)**

macOS Golden Gate soll adaptive Steuerelemente für MacBook-Pro-Modelle mit Touchscreen einführen. Controls sollen je nach Eingabemethode (Touch/Maus) adaptieren.

---

## **Migration**

### **Apple Silicon Pflicht** **Bestätigt**

macOS Golden Gate unterstützt ausschließlich Macs mit Apple Silicon (M-Chip). macOS 26 Tahoe war das letzte Release mit Intel-Unterstützung.

**Nicht mehr unterstützte  
Intel-Mac-Modelle ab  
macOS 27:**

Modell

Chip

Letztes macOS

MacBook Pro 13" (2020, 4x Thunderbolt 3)

Intel

macOS 26 Tahoe

MacBook Pro 16" (2019)

Intel

macOS 26 Tahoe

iMac 27" (2020)

Intel

macOS 26 Tahoe

Mac Pro (2019)

Intel

macOS 26 Tahoe

**Folgen für Entwickler:** - Intel-only Targets können entfernt werden - Universal Binaries können auf Apple Silicon reduziert werden - CI/CD-Pipeline auf Apple Silicon prüfen

### **Rosetta 2 - Letzte vollständige Unterstützung** **Bestätigt**

**macOS 27 Golden Gate ist die letzte Version, die Rosetta 2 vollständig unterstützt.**

**macOS 27 Golden Gate** (Herbst 2026): Rosetta 2 vollständig verfügbar

**macOS 28** (Herbst 2027): Rosetta 2 wird weitgehend eingestellt


**Handlungsempfehlung:** - Intel-abhängige Drittbibliotheken jetzt identifizieren - Migration auf native Apple-Silicon-Builds bis spätestens macOS-28-Cycle planen - Deadline: vor macOS 28 (Herbst 2027)


Quelle: Apple (bestätigt), TechRepublic, TechTimes

---

## watchOS 27

### Beta 1

**Datum:** 08. Juni 2026  Bestätigt

**Build-Nummer:** 24R5289n  Bestätigt

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), WWDC 2026 Keynote

**Neues Zifferblatt: Modular Ultra Variante**  Bestätigt

Eine neue Variante des „Modular Ultra“-Zifferblatts ist jetzt auch für Nicht-Ultra-Apple-Watch-Modelle verfügbar: - Große Zeitanzeige im oberen Bereich - Drei Komplikationen darunter

**Quelle:** MacRumors WWDC 2026 Coverage

**Herzfrequenz-Tracking**  Bestätigt

Verbesserte Herzfrequenz-Messung.

**Quelle:** MacRumors

**Siri / Apple Intelligence**  Bestätigt

Gleiche neue Siri-Fähigkeiten wie auf iOS

Cross-Platform Siri: Chat-Synchronisation via iCloud zwischen Watch, iPhone, Mac

**Writing Tools**  Bestätigt

Grammatikprüfung in Writing Tools auch auf watchOS 27.

**Performance-Fokus**  Bestätigt

Allgemeiner Fokus auf Stabilität und Performance (Snow-Leopard-Ansatz).

---

## Migration

Keine spezifischen Breaking Changes für watchOS 27 bekannt. Apple-Intelligence- und Siri-Änderungen (App Intents) gelten analog.

---

## visionOS 27

### Beta 1

**Datum:** 08. Juni 2026  Bestätigt

**Quelle:** WWDC 2026 Keynote

## Wheelchair Control via Eye-Tracking Bestätigt

**Quelle:** Apple Accessibility Announcement, 19. Mai 2026  
visionOS 27 führt Wheelchair Control ein: Rollstuhlnutzer können Rollstühle mit assistiver Technologie durch Eye-Tracking und Headset-Steuerung kontrollieren.

## Siri / Apple Intelligence Bestätigt

Alle neuen Siri- und Apple-Intelligence-Features auch auf visionOS 27.

## Design-Anpassungen Bestätigt

Liquid-Glass-Überarbeitungen konform zu iOS 27 und macOS Golden Gate.

---

## Migration

Keine spezifischen Breaking Changes für visionOS 27 bekannt.

---

## SwiftUI

### Neue Features

#### Adaptive Layout APIs Bestätigt

Neue SwiftUI-Komponenten für faltbare Formfaktoren (iPhone Fold): - Hinge-State-Detection in SwiftUI - Multi-Configuration-Screen-Handling - Erweiterte NavigationSplitView für adaptive Displays

**Quelle:** WWDC 2026 (TechTimes, byteiota.com)

#### Liquid Glass Compositing Bestätigt

Das `.glassEffect()` Modifier-System wird mit Updates zum Compositing-Pipeline ausgeliefert: - Propagiert automatisch auf alle Apps mit `GlassEffectContainer` - Neue Intensity-Parameter für den Opacity-Slider (systemgesteuert)

#### Design Tokens - zweite Iteration Bestätigt

Liquid Glass Design Language erhält zweite Iteration: - Aktualisierte Design Tokens - Neue Material Guidelines im SDK - Tighter Corner Radius als Standard

**Quelle:** WWDC 2026 (TechTimes Developer Coverage)

#### Toolbar-APIs - neue Modifier Bestätigt

**Quelle:** [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

*// Sichtbarkeitspriorität in anpassbaren Toolbars*

```
.visibilityPriority(.high)
```

*// Niedrig-priorisierte Items dauerhaft ins Overflow-Menü*

```
.toolbarOverflowMenu { /* items */ }
```

*// Kritische Aktionen am rechten Rand fixieren*

```
.topBarPinnedTrailing { ShareButton() }
```

// Navigation Bar automatisch beim Scrollen minimieren

```
.toolbarMinimizeBehavior(.automatic)
```

### Document API Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

Neue Protokolle für direkte Dokumentensteuerung: - WritableDocument – async, inkrementelle Schreiboperationen auf Disk - ReadableDocument – asynchrones Lesen mit Fortschrittsanzeige via Foundation Subprogress API - DocumentCreationSource – mehrere Erstellungsquellen mit je eigenem NewDocumentButton

### Reorderable Containers Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

Drag-to-Reorder jetzt in jedem Container (nicht nur List): - Gleicher Code für List, LazyVGrid und Custom Layouts - Erstmals verfügbar auf watchOS - Neuer Modifier `swipeActionsContainer` für Swipe-Aktionen in beliebigen ScrollViews

### Dialog/Alert - Item-Binding Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

Confirmation Dialogs und Alerts unterstützen jetzt Item-Binding-Pattern (analog zu Sheets):

```
.confirmationDialog(item: $selectedItem) { item in  
    Button("Löschen") { delete(item) }  
}
```

### AsyncImage - HTTP-Caching Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

AsyncImage respektiert jetzt standardmäßig Server-Cache-Header. Eigener URLSession mit URLCache via `asyncImageURLSession`-Modifier konfigurierbar.

### @State als Macro mit Lazy-Initialisierung Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

@State ist jetzt ein Macro mit Lazy-Initialisierung für Klassen. Klassen werden nur einmal pro View-Lifetime initialisiert.

### Build-Time-Verbesserungen via ContentBuilder Bestätigt

Quelle: [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/)

ViewBuilder ist jetzt als ContentBuilder exponiert. Ermöglicht signifikante Build-Zeit-Verbesserungen in Xcode 27.

---

### Spatial Preview Framework Bestätigt (Platforms State of the Union)

Quelle: WWDC 2026 Platforms State of the Union (09. Juni 2026)

Das neue **Spatial Preview Framework** ermöglicht es Mac-Apps, 3D-Modelle um Vision-Pro-Nutzer in Echtzeit zu erweitern (Spatial Computing auf macOS):

```
import SpatialPreview
```

// Mac-App erweitert 3D-Modell für Vision-Pro-Nutzer

```
let preview = SpatialPreviewSession()
try await preview.present(model: myARModel, aroundUser: true)
```

**Relevanz für Entwickler:** - Mac-Apps können 3D-Previews direkt an Vision Pro senden - Kein visionOS-Build erforderlich — läuft über macOS-App - Ideal für 3D-Design-, CAD- und Produktvisualisierungsapps

---

## Breaking Changes

*actionSheet deprecated*  **Bestätigt**

*// Nicht mehr verwenden*

```
.actionSheet(isPresented: $showingActionSheet) {
    ActionSheet(title: Text("Aktion wählen"), buttons: [...])
}
```

*// Verwenden*

```
.confirmationDialog("Aktion wählen", isPresented: $showingActionSheet, titleVisibility: .visible)
{
    Button("Option A") {}
    Button("Abbrechen", role: .cancel) {}
}
```

*GlassEffectContainer Grenzen prüfen*  **Bestätigt**

In Mixed UIKit/SwiftUI-Apps: GlassEffectContainer-Grenzen auditieren, um konsistentes Morphing-Verhalten über beide Frameworks sicherzustellen.

---

## Bekannte Probleme

Bekannte SwiftUI-Probleme aus Beta-1-Testing werden nach offizieller Apple-Dokumentation ergänzt.

---

## Beispiele

*// Adaptive Layout für faltbare Displays*

```
struct AdaptiveContentView: View {
    @Environment(\.horizontalSizeClass) var sizeClass

    var body: some View {
        if sizeClass == .regular {
            NavigationSplitView {
                SidebarView()
            } detail: {
                DetailView()
            }
        } else {
            NavigationStack {
                ContentView()
            }
        }
    }
}
```

```
}  
}  
}
```

---

## Swift 6.4

**Swift-Version bestätigt:** Swift 6.4 ist in Xcode 27 Beta 1 (Build 27A5194q) enthalten. Quelle: [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/) – 08. Juni 2026. Aktualisiert am 09. Juni 2026 mit Platforms-State-of-the-Union-Erkenntnissen.

### Neue Features

**OS-Kernel-Teile in Swift**  **Bestätigt (Platforms State of the Union)**

**Quelle:** WWDC 2026 Platforms State of the Union (09. Juni 2026)

Apple schreibt erstmals Teile des Betriebssystem-Kernels in Swift. Dies unterstreicht die Eignung von Swift für systemnahe Programmierung und ist ein bedeutendes Bekenntnis zu Swift als Systemsprache.

**Relevanz für Entwickler:** - Swift ist jetzt offiziell für systemnahe und performance-kritische Bereiche bestätigt - Weitere Swift-Systemframework-Migrationen in künftigen Releases wahrscheinlich

**Vereinfachte Availability-Syntax: anyAppleOS**  **Bestätigt**

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

*// Vorher (Swift 6.3 und älter)*

```
@available(iOS 27, macOS 27, watchOS 27, tvOS 27, visionOS 27, *)
```

```
func myFunction() {}
```

*// Nachher (Swift 6.4)*

```
@available(anyAppleOS 27, *)
```

```
func myFunction() {}
```

**Unterdrückbare Compiler-Warnungen**  **Bestätigt (Platforms State of the Union)**

**Quelle:** WWDC 2026 Platforms State of the Union (09. Juni 2026), [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

Swift 6.4 führt **Suppressible Compiler Warnings** ein: Warnungen können auf Callsite- oder Methodenebene gezielt unterdrückt werden. Ergänzt wird dies durch verbesserte Type-Checker-Diagnostics für präzisere Fehlermeldungen.

**@diagnose-Attribut**  **Bestätigt**

Feingranularere Kontrolle über Compiler-Warnungen auf Methodenebene.

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

**Async-Unterstützung in defer-Statements**  **Bestätigt**

defer-Blöcke können jetzt asynchrone Operationen enthalten. Cleanup-Code läuft garantiert – unabhängig davon, ob eine Funktion regulär zurückkehrt oder einen Fehler wirft.

```
func processData() async throws {
```

```
    defer {
```

```
        await cleanup() // NEU: async in defer erlaubt
```

```
}  
try await doWork()  
}
```

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

### **Neues Iteration-Protokoll für Non-Copyable-Types** **Bestätigt**

For-in-Schleifen funktionieren jetzt mit Non-Copyable-Types wie Span und InlineArray ohne unnötiges Kopieren.

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

### **Foundation Performance-Verbesserungen** **Bestätigt**

URL-Parsing bis zu **4x schneller** in Swift 6.4.

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

### **Swift Testing XCTest-Interoperabilität** **Bestätigt**

Verbesserte schrittweise Migration zwischen Swift Testing und XCTest-Frameworks möglich.

**Quelle:** [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/)

---

## Concurrency

Das Concurrency-Modell erhält Refinements in Swift 6.4 / iOS 27: - Stärkere Data-Isolation-Garantien - Weniger nötige @Sendable und @MainActor-Annotierungen durch verbesserte Inferenz - Verbesserter async/await-Compiler-Output - Async-Unterstützung in defer-Statements (neu in 6.4)

---

## Breaking Changes

### **@available-Syntax (kein Breaking Change - rückwärtskompatibel)**

anyAppleOS ist additiv – bestehender Code mit einzelnen Plattform-Angaben kompiliert weiterhin ohne Anpassung.


Bekanntere weitere Breaking Changes werden nach Veröffentlichung der offiziellen Swift 6.4 Release Notes auf [swift.org](https://swift.org) ergänzt.

---

## Xcode 27

### IDE

**Beta 1:** 08. Juni 2026  **Bestätigt**

**Build-Nummer:** 27A5194q  **Bestätigt**

**Enthaltene Swift-Version:** Swift 6.4  **Bestätigt**

**Quelle:** [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/), WWDC 2026, WWDC 2026 Platforms State of the Union (09. Juni 2026)

### **Allgemeine IDE-Änderungen** **Bestätigt (Platforms State of the Union)**

**Quelle:** WWDC 2026 Platforms State of the Union, MacRumors, TechTimes (09. Juni 2026)

**30% kleiner:** Xcode 27 ist 30 % kleiner als Xcode 26

**Ausschließlich Apple Silicon:** Intel-Macs werden nicht mehr unterstützt

**iCloud-Einstellungssynchronisierung:** Xcode-Einstellungen werden über iCloud auf alle Macs synchronisiert

**Anpassbare Toolbar:** Vollständig konfigurierbare Toolbar

**Per-Projekt-Themes:** Farbschemata pro Projekt konfigurierbar (Farben übernehmen den gesamten Editor)

**Xcode Cloud 2x schneller:** Metal- und visionOS-Builds werden auf Apple Silicon ausgeführt

**Relevanz für Entwickler:** - Xcode 27 setzt macOS Golden Gate (Apple Silicon) voraus — CI/CD auf Apple Silicon prüfen - Kleinere App-Größe beschleunigt CI/CD-Pipelines

**On-Device KI Code Completion**  **Bestätigt (Platforms State of the Union)**

**Quelle:** TechTimes (09. Juni 2026), [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/)

**Vollständig On-Device:** Läuft auf Apple Silicons Neural Engine — kein Quellcode verlässt den Mac

**Privacy-First-Architektur:** GitHub Copilot und Cursor können das nicht bieten

**Lokal + Cloud:** Schwerere Anfragen werden wahlweise an externe Modelle (Anthropic, Google, OpenAI) delegiert

Inline Code Suggestions, Dokumentation schreiben, Issues beheben direkt im Editor

**Coding Agents**  **Bestätigt (WWDC 2026)**

**Quelle:** [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/)

Multi-Modell-Unterstützung mit anpassbaren KI-Assistenten (Anthropic, Google, OpenAI integriert)

Agenten können mit Simulator interagieren, Lokalisierungen einrichten, Tests ausführen, Abstürze beheben

**/plan-Befehl:** Arbeit vorab scopen, bevor Code geändert wird

Agent-Konversationen direkt im Editor (Tab und Split-View unterstützt)

Intelligente Aufgabenverteilung: Prototyp-Generierung, Implementation Details, Polishing

Solo- und Team-Workflows werden unterstützt

**Device Hub**  **Bestätigt (WWDC 2026)**

**Quelle:** [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/)

Ersetzt das klassische **Simulator**-Fenster vollständig

Zentrale Verwaltung aller Geräte und Simulatoren an einem Ort

Probleme diagnostizieren und reproduzieren ohne IDE-Verlassen

Gerätezustand direkt inspizieren

Accessibility-Einstellungen und iPhone-Mirroring-Resize-Tests integriert

**Icon Composer**  **Bestätigt (bereits in Xcode 26)**

Mehrschichtige Icons mit Liquid Glass

Dynamische Beleuchtungsvorschau für iPhone, iPad, Mac und Apple Watch

## Lokalisierung Bestätigt

Agenten richten String Catalogs ein und generieren Übersetzungen für mehrere Sprachen automatisch  
Verbesserte Lokalisierungskatalog-Features

## Organizer-Neuerungen Bestätigt (Platforms State of the Union)

**Quelle:** WWDC 2026 Platforms State of the Union

Hochpriorisierte Probleme werden zuerst angezeigt

Neue Metriken: Storage und Animation Hitches

**Metric Goals:** Eigene Performance-Ziele definieren

KI-gestützte Fix-Empfehlungen (Agent-powered)

## Liquid Glass Kompatibilitäts-Flag entfernt Bestätigt

Das Build-Setting `UIDesignRequiresCompatibility = true` wird in Xcode 27 vollständig entfernt. Alle gegen das iOS-27-SDK gebauten Apps erhalten automatisch Liquid Glass.

---

## Simulator

Der klassische Simulator wurde durch den **Device Hub** ersetzt (siehe IDE-Abschnitt). Alle Simulator-Funktionen sind weiterhin verfügbar — jetzt zentral im Device Hub integriert.

---

## Debugging

### Instruments Bestätigt (WWDC 2026)

**Quelle:** [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/)

**Swift Concurrency Instrument:** Visibility in Async-Task-Scheduling, Actor-Contention und Thread-Usage

**Time Profiler:** CPU-Bottleneck-Identifikation

**System Trace:** Tieferer System-Level-Blick auf App-Interaktionen

**Top Functions:** Schwere Call-Stacks schnell identifizieren

**Run Comparisons:** Parallele Impact-Messung (Side-by-Side)

SwiftUI Instrument: Visualisierung von Datenänderungen und deren Auswirkungen auf View-Updates

---

## Testing

### Swift Testing Erweiterungen Bestätigt (WWDC 2026)

Swift Testing erhält Erweiterungen für verbesserte Test-Workflows.

### XCUIAutomation Tests Bestätigt (bereits in Xcode 26, weiter verbessert)

Record, run und maintain UI-Tests direkt in Xcode

Test Replay über Lokalisierungen, Gerätetypen und Systemzustände

---

# Apple Intelligence

## Bestätigte Features (WWDC 2026)

**Quellen:** WWDC 2026 Keynote, 9to5Mac, Engadget, MacRumors

### *Zweite-Gen On-Device Modell*

Neues Second-Gen On-Device Modell

Personal Context Understanding: Spotlight-Semantic-Index für bessere Suche und Awareness

Weltweites Wissen (Broad World Knowledge)

App-Aktionen (App Actions) und On-Screen Awareness

### *Foundation Models Framework – Erweiterungen*

**Multimodaler Input:** Third-Party-Apps können jetzt Bilder (zusätzlich zu Text) an das On-Device-Modell übergeben

**Externe Modelle:** Entwickler können andere Modelle (neben Apple Foundation Models) in ihre Apps einbinden

**Weiterhin:** Text-Extraktion, Summarisierung, Guided Generation, Tool Calling, vollständig On-Device

**Technischer Hintergrund:** Apple Foundation Models 3 (AFM 3) ist ein Sparse-MoE-Modell mit ~20 Milliarden Gesamtparametern, von denen pro Prompt nur 1–4 Milliarden aktiv sind. Die Vision-Komponente nutzt kontrastives Pretraining (CLIP-Stil). Das Vorgängermodell (AFM 2 / iOS 26) nutzte ein ~3B-Parameter-Vollmodell — AFM 3 ist deutlich leistungsfähiger bei ähnlichem Energieverbrauch.

### *Image Playground Bestätigt*

Bilder in nahezu beliebigen Stilen generieren, inklusive realistischerer Optionen

Plattformübergreifende Integration auf iOS, iPadOS, macOS

Nahezu beliebige Bildgrößen

ChatGPT-Integration für zusätzliche Stile

### *Siri AI (Apple Intelligence-Ebene) Bestätigt*

Personal Context: E-Mail, Kalender, Fotos, Kontakte

On-Screen Awareness

App Actions: Direktaktionen in Apps via Siri

### *Photos KI-Bearbeitung Bestätigt*

Clean Up überarbeitet

Extend: KI-generierte Bilderweiterungen über Ränder hinaus

Reframe: Perspektivänderung bei räumlichen Fotos

### *Writing Tools Bestätigt*

Grammatikprüfung system-weit

Proofreading in Drittanbieter-Apps via Voice Dictation

### *Accessibility AI-Features Bestätigt*

**Quelle:** Apple Accessibility Announcement, 19. Mai 2026

VoiceOver Image Explorer: Erweiterter KI-Bildbeschreibungsmodus

Magnifier: Visuelle Beschreibungen und verbesserte Erkennung

Voice Control: Natural Language Support

Accessibility Reader: Verbesserte Darstellung komplexer Layouts

Generated Subtitles: On-Device Transkription (verbessert)

FaceTime Sign Language API: Neue API für Gebärdensprach-Interpretation

Name Recognition: Ausweitung auf 50+ Sprachen

visionOS: Wheelchair Control via Eye-Tracking

### Siri Extensions API für Drittanbieter-KI Bestätigt

Apple öffnet Siri für alternative KI-Chatbots: - Google Gemini - Anthropic Claude - OpenAI ChatGPT

Nutzer wählen ihren bevorzugten KI-Dienst in Einstellungen > Extensions. Ein dedizierter App-Store-Bereich für Extensions-kompatible Apps wird eingerichtet.

### Core AI Framework (Core ML Nachfolger) Bestätigt

Ersetzt Core ML als primäres KI-Framework

Standardisierter Plugin-Mechanismus via Model Context Protocol (MCP)


Beliebige MCP-kompatible Modelle integrierbar

Core ML: Deprecated, aber noch funktionsfähig (Übergangszeit)

---

## Apple KI-Entwicklung für Entwickler

**Stand:** iOS 27 Beta 1 – 08. Juni 2026

 Beta-APIs – Breaking Changes zwischen Betas möglich. Dieses Kapitel wird bei jeder Beta mit Warnhinweisen aktualisiert.

**Dieses Kapitel** ist ein dauerhaftes Grundlagenkapitel aus reiner Entwicklerperspektive. Es erklärt alle KI-Frameworks und APIs in iOS 27, was Entwickler implementieren müssen, und wie Apps für Apples KI-Systeme zugänglich gemacht werden.

---

## Architektur-Übersicht

Apple hat in iOS 27 eine dreischichtige KI-Architektur eingeführt:

Schicht

Framework

Ausführungsort

Datenschutz

On-Device

Foundation Models Framework

Lokal auf Gerät

Vollständig privat

Private Cloud Compute

Apple Server Models

Apple-Server

Kryptografisch gesichert, kein Apple-Zugriff

Third-Party AI

Siri Extensions API

Externer Anbieter

Nutzer-kontrolliert

ML & Custom Models

Core AI Framework

On-Device + extern via MCP

Konfigurierbar

System Intelligence

Apple Intelligence

On-Device + Private Cloud

Systemlevel

## Welches Framework für welchen Zweck?

Ziel

Framework / API

Text generieren, zusammenfassen, analysieren

Foundation Models Framework

Eigenes ML-Modell einbinden

Core AI Framework

App von Siri verstehen und steuern lassen

App Intents (Pflicht)

Eigene App als Siri-KI-Anbieter registrieren

Siri Extensions API

Schreibwerkzeuge im eigenen Textfeld

Writing Tools API

KI-Bilder in eigener App generieren

Image Playground API

SiriKit-Apps migrieren


App Intents (SiriKit deprecated)

Core-ML-Apps migrieren

Core AI Framework (Core ML deprecated)

### Foundation Models Framework

Das **Foundation Models Framework** ist das wichtigste neue KI-Framework für App-Entwickler in iOS 27. Es bindet Apples On-Device-Sprachmodell direkt in eigene Apps ein — vollständig lokal, keine Daten verlassen das Gerät.

**Hinweis:** Apple plant, das Foundation Models Framework **noch im Sommer 2026 als Open Source** zu veröffentlichen.  Bestätigt (Platforms State of the Union, 09. Juni 2026)

**Apple Foundation Model 3 (AFM 3)**  **Bestätigt (Platforms State of the Union)**

**Quelle:** WWDC 2026 Platforms State of the Union, ofox.ai (09. Juni 2026)

## Das neue On-Device-Modell in iOS 27:

Eigenschaft

Wert

Architektur

Sparse (Mixture of Experts)

Parameter gesamt

~20 Milliarden

Aktive Parameter pro Prompt

1-4 Milliarden

Vision-Pretraining

Kontrastiv (CLIP-Stil)

Name

AFM 3 Core Advanced

**Private Cloud Compute - kostenloses Kontingent:** - Entwickler mit **weniger als 2 Millionen Erstdownloads** im App Store erhalten kostenlosen Zugang zu Private Cloud Compute - Komplexere Anfragen werden automatisch auf Apple-Silicon-Servern ausgeführt — ohne Datenspeicherung - **Quelle:** Platforms State of the Union (09. Juni 2026)

## Neue Features in iOS 27 (zusätzlich zu iOS 26):

Feature

Status

Quelle

Multimodaler Input (Bilder + Text)

Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Externe Modelle (Claude, Gemini) via Language Model Protocol

✅ Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Vision Framework Tools (OCR, Barcode) für Modelle

✅ Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Dynamic Profiles (Modell, Tools, Instructions on-the-fly)

✅ Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Private Cloud Compute (kostenlos für Small Business <2M Downloads)

✅ Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Evaluations Framework (KI-Features bei dynamischen Bedingungen testen)

✅ Bestätigt

[developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

## Verfügbarkeit:

Plattform

Min. Version

Min. Chip

Einschränkung

iOS

27.0

A17 Pro (iPhone 15 Pro)

Ältere Chips: reduzierte Modellkapazität

iPadOS

27.0

M1

-

macOS

Golden Gate

M1

-  
visionOS

27.0

M2

-

## import FoundationModels

### Grundlegende Textgenerierung

```
let session = LanguageModelSession()
```

```
let response = try await session.respond(to: "Erkläre Swift Concurrency in einem Satz.")  
print(response.content)
```

### System-Prompt (Kontext setzen)

```
let session = LanguageModelSession(  
    instructions: "Du bist ein hilfreicher Assistent für MeineApp. Antworte immer auf Deutsch und  
    kurz."  
)
```

### Multi-Turn Konversation

```
let session = LanguageModelSession()
```

```
let response1 = try await session.respond(to: "Mein Name ist Max.")
```

```
let response2 = try await session.respond(to: "Wie heiße ich?")
```

```
// response2.content enthält "Max" — Kontext bleibt über mehrere Anfragen erhalten
```

### Streaming-Antworten

```
let session = LanguageModelSession()
```

```
let stream = session.streamResponse(to: prompt)
```

```
for try await partial in stream {  
    await MainActor.run {  
        textView.text = partial.content  
    }  
}
```

### Guided Generation - Strukturierte Ausgabe Experimentell (Beta 1)

@Generable erzwingt typsichere, strukturierte Ausgaben — kein manuelles JSON-Parsing:

```
@Generable
```

```
struct ProductAnalysis {
```

```
    @Guide(description: "Produktname")
```

```
    var name: String
```

```
    @Guide(description: "Bewertung von 1 bis 5")
```

```
    var rating: Int
```

```

@Guide(description: "Kurze Zusammenfassung in einem Satz")
var summary: String
}

```

```

let session = LanguageModelSession()
let analysis = try await session.respond(
  to: "Analysiere: \((productDescription))",
  generating: ProductAnalysis.self
)
// analysis.name, analysis.rating, analysis.summary — typsicher, nie nil

```

### Tool Calling - Eigene Funktionen bereitstellen Experimentell (Beta 1)

Das Modell kann eigene Swift-Funktionen aufrufen:

```

struct WeatherTool: Tool {
  static let name = "getCurrentWeather"
  static let description = "Gibt das aktuelle Wetter für eine Stadt zurück"

  @Parameter(description: "Stadtname")
  var city: String

  func call() async throws -> ToolOutput {
    let weather = try await WeatherService.fetch(for: city)
    return ToolOutput(weather.description)
  }
}

```

```

let session = LanguageModelSession(tools: [WeatherTool()])
let response = try await session.respond(to: "Wie ist das Wetter in München?")

```

### Multimodaler Input - Bilder (neu in iOS 27) Bestätigt

```

let session = LanguageModelSession()
let image = UIImage(named: "product_photo")!

let prompt = LanguageModelPrompt(
  text: "Was siehst du auf diesem Bild? Beschreibe es kurz.",
  images: [image]
)

```

```

let response = try await session.respond(to: prompt)

```

### Dynamic Profiles (neu in iOS 27) Bestätigt

Quelle: [developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/)

Modell, Tools und Instructions können innerhalb einer Session dynamisch gewechselt werden:

```

let session = LanguageModelSession()

// Profil für Textzusammenfassung
try await session.setProfile(.summarization)

```

```
let summary = try await session.respond(to: "Fasse folgendes zusammen: \(\longText)")
```

```
// Profil wechseln für Code-Analyse
```

```
try await session.setProfile(.codeAnalysis)
```

```
let codeReview = try await session.respond(to: "Analysiere: \(\codeSnippet)")
```

*LanguageModel-Protokoll - Anbieter ohne Code-Änderungen wechseln*  *Bestätigt (Platforms State of the Union)*

**Quelle:** TechTimes, MacRumors, Platforms State of the Union (09. Juni 2026)

Das LanguageModel-Protokoll ist eine öffentliche Swift-Schnittstelle, die es erlaubt, KI-Anbieter zu wechseln, **ohne die Session-Logik oder den restlichen App-Code ändern zu müssen**. Entwickler können z. B. mit Apples On-Device-Modell prototypen und komplexe Abfragen zu Gemini oder Claude routen — allein durch Änderung einer Swift-Package-Manager-Abhängigkeit.

Drittanbieter-Modelle (z. B. Claude, Gemini) können über das LanguageModel-Protokoll eingebunden werden:

```
// Eigenes Modell  
als  
LanguageModel-  
Implementierung  
class  
MyCustomLanguage  
eModel:  
LanguageModel {  
    func respond(to  
prompt:  
LanguageModelPr  
ompt) async  
throws ->  
LanguageModelRe  
sponse {  
    // Eigene  
Modell-Inferenz  
    }  
}  
  
let session =  
LanguageModelSes  
sion(model:  
MyCustomLanguage  
eModel())
```

**Drei-Tier-Routing:**

Tier

Ausführungsort

Wann

On-Device

Neural Engine

Routineabfragen

Private Cloud Compute


Apple-Silicon-Server

Komplexe Anfragen

Cloud-Routing

Drittanbieter-Cloud (Gemini / Claude / ChatGPT)

Schwerste Workloads

**Eingebaute Tools:** *BarcodeReaderTool, OCRTool, Spotlight-Suche*  *Bestätigt (Platforms State of the Union)*

**Quelle:** TechTimes (09. Juni 2026)

Apple hat neue Built-In-Tools für Foundation Models eingeführt, die vom Vision Framework angetrieben werden:

```
let session = LanguageModelSession(tools: [  
    BarcodeReaderTool(), // Barcodes aus Bildern lesen  
    OCRTool(),          // Text aus Bildern extrahieren (via Vision)  
    SpotlightSearchTool() // Vollständig lokales RAG via Spotlight-Index  
)
```

```
let response = try await session.respond(  
    to: "Lies den Barcode auf diesem Foto und erkläre das Produkt.",  
    images: [productImage]  
)
```

**SpotlightSearchTool:** Ermöglicht vollständig lokales **Retrieval-Augmented Generation (RAG)** — zuvor mussten Entwickler eine eigene Vektordatenbank-Infrastruktur aufbauen und pflegen.

**Verfügbarkeit prüfen & Graceful Degradation**

Immer Fallback implementieren — Foundation Models ist nicht auf allen iOS-27-Geräten vollständig verfügbar:

```
func analyzeText(_ text: String) async {  
    guard LanguageModelSession.isAvailable else {  
        showFallbackMessage()  
        return  
    }  
  
    do {  
        let session = LanguageModelSession()  
        let response = try await session.respond(to: text)
```

```
    showResult(response.content)
} catch LanguageModelError.modelUnavailable {
    showFallbackMessage()
} catch LanguageModelError.contextLengthExceeded {
    await analyzeText(String(text.prefix(2000)))
} catch {
    showError(error)
}
}
```

### **Datenschutz**

Läuft vollständig **On-Device** — keine Daten verlassen das Gerät

Private Cloud Compute wird automatisch genutzt, wenn die Anfrage zu komplex ist

Via **Per-Intent Privacy Declarations** (App Intents) kann Cloud-Routing explizit verboten werden


---

### **Core AI Framework**

Das **Core AI Framework** ersetzt Core ML als primäres ML-Framework.

**Status Core ML:** Deprecated in iOS 27. Bestehende Implementierungen laufen weiterhin.  
Migrationsfenster: ca. 2–3 Jahre.

**Verfügbarkeit:** iOS 27+, iPadOS 27+, macOS Golden Gate+, A13+

**Neue Fähigkeiten (Platforms State of the Union) **  
**Bestätigt**

**Quelle:** WWDC 2026 Platforms State of the Union (09. Juni 2026)

Feature

Beschreibung

Ahead-of-Time-Kompilierung

Custom-Modelle werden bei Build-Zeit kompiliert — schnellere Inferenz

Dedizierte Instruments

Spezialisierte Profiling-Tools für Core-AI-Modelle in Xcode Instruments

Siri-Power

Core AI Framework treibt Siris On-Device-Verarbeitung an

Python-Tools

PyTorch-Modelle direkt zu Apple Silicon konvertieren

## PyTorch → Apple Silicon Konvertierung (neu)

# Python-Tool (neu in Core AI Framework / WWDC 2026)

```
import coreai_tools
```

```
model = coreai_tools.convert_from_pytorch(  
    model=my_pytorch_model,  
    target="apple_silicon",  
    quantization="int4"  
)  
model.save("MyModel.coreai")
```

```
import CoreAI
```

## Unterschied zu Core ML

Merkmal

Core ML

Core AI Framework

Status

Deprecated (iOS 27)

Aktuell (iOS 27+)

Externe Modelle

Nein

Ja — via MCP

Plugin-Mechanismus

Nein

Ja

Async/Await

Eingeschränkt

Vollständig

MCP-Support

Nein

Ja

### Modell laden und verwenden

```
import CoreAI
```

```
let model = try await AIModel.load(from: modelURL)  
let prediction = try await model.predict(input: myInput)
```

### MCP-Integration - Externes Modell einbinden Experimentell (Beta 1)

```
let connector = MCPConnector(  
  endpoint: URL(string: "https://your-model-server.com/mcp")!  
)  
let model = try await AIModel.load(via: connector)  
let result = try await model.predict(input: myInput)
```

### Plugin-Mechanismus

```
// Plugin registrieren (z.B. in App init)
```

```
CoreAI.registerPlugin(MyCustomModelPlugin())
```

```
class MyCustomModelPlugin: AIModelPlugin {  
  var identifier: String { "com.myapp.custommodelplugin" }  
  
  func predict(input: AIInput) async throws -> AIOutput {  
    // Eigene Modell-Inferenz  
  }  
}
```

### Migration von Core ML

```
// — Vorher: Core ML
```

---

```
import CoreML
```

```
let model = try MLModel(contentsOf: modelURL)  
let prediction = try model.prediction(from: input)
```

```
// — Nachher: Core AI Framework
```

---

```
import CoreAI
```

```
let model = try await AIModel.load(from: modelURL)  
let prediction = try await model.predict(input: input)
```

---

## Apple Intelligence Integration

**Apple Intelligence** ist Apples systemweites KI-Layer — kein einzelnes Framework, sondern ein Sammelbegriff. Als Entwickler greift man über die konkreten Frameworks darunter zu.

Apple Intelligence Komponente

Entwickler-Zugang

On-Device Sprachmodell

Foundation Models Framework

Personal Context (Kontakte, Mail, Kalender)

App Intents + Siri

Writing Tools

Writing Tools API (automatisch in UITextView)

Image Playground

Image Playground API

Visual Intelligence

Vision Framework

Siri On-Screen Awareness

App Intents

App Actions via Siri

App Intents

**Was passiert automatisch ohne Code-Änderungen?** - Writing Tools erscheinen in jedem UITextView und UITextView - Siri erkennt App-Namen und einfache Navigation

**Was erfordert aktive Implementierung?** - Siri soll App-spezifische Aktionen ausführen → **App Intents** (Pflicht) - KI-Text-Features in eigener App → **Foundation Models Framework** - Bildgenerierung → **Image Playground API** - Eigene App als Siri-KI-Anbieter → **Siri Extensions API**

---

## Siri & App Intents (Pflicht-Integration)

App Intents sind die **einzige** Methode, um iOS-27-Siri-Features in eigene Apps zu integrieren.

**SiriKit ist deprecated** seit WWDC 2026. Migrationsfenster: ca. 2-3 Jahre.

**Quelle Deprecation:** WWDC 2026 Keynote, [techtimes.com/articles/318005](https://techtimes.com/articles/318005) (08. Juni 2026)

**Neue App-Intents-Fähigkeiten in iOS 27:** - **Streaming-Antworten:** Mehrstufiger Dialog bleibt in einem App-Kontext (kein generischer Ergebnis-Screen mehr) - **Reichhaltigere Entity-Typen:** Komplexere Datenstrukturen für App-Interaktionen - **Strukturiertes Konversations-Follow-up:** Siri hält Gesprächsfluss über mehrere Exchanges aufrecht - **Entity-Schemas:** App-Inhalte im Spotlight Semantic Index bereitstellen  Bestätigt (Platforms State of the Union) - **Intent-Schemas:** Natural-Language-Aktionen ohne hartkodierte Phrasen  Bestätigt (Platforms State of the Union) - **View Annotations API:** Views auf Entities mappen — Siri kann mit sichtbaren Inhalten konversationell interagieren  Bestätigt (Platforms State of the Union) - **App Intents Testing Framework:** Integration über echte Systempfade (Siri, Shortcuts, Spotlight) testen  Bestätigt

### Minimal-Integration

#### import AppIntents

```
struct OpenFavoritesIntent: AppIntent {
    static var title: LocalizedStringResource = "Favoriten öffnen"
    static var description = IntentDescription(
        "Öffnet die Favoritenliste in MeineApp"
    )

    func perform() async throws -> some IntentResult {
        await MainActor.run { AppNavigator.openFavorites() }
        return .result()
    }
}
```

### App Shortcuts - automatisch in Siri, Spotlight und Kurzbefehle

```
struct MeineAppShortcuts: AppShortcutsProvider {
    static var appShortcuts: [AppShortcut] {
        AppShortcut(
            intent: OpenFavoritesIntent(),
            phrases: [
                "Öffne Favoriten in \(.applicationName)",
                "Zeige meine Favoriten"
            ],
            shortTitle: "Favoriten",
            systemImageName: "star.fill"
        )
    }
}
```

### Parameter & Entities

```
struct FindProductIntent: AppIntent {
    static var title: LocalizedStringResource = "Produkt suchen"
```

```

@Parameter(title: "Produktname")
var productName: String

func perform() async throws -> some IntentResult & ReturnsValue<ProductEntity> {
    let product = try await ProductService.find(name: productName)
    return .result(value: ProductEntity(product))
}
}

```

```

struct ProductEntity: AppEntity {
    static var typeDisplayRepresentation: TypeDisplayRepresentation = "Produkt"
    static var defaultQuery = ProductQuery()

    var id: String
    var name: String

    var displayRepresentation: DisplayRepresentation {
        DisplayRepresentation(title: "\ (name)")
    }
}

```

Streaming-Antworten in App Intents (neu in iOS 27) 🚧 Experimentell (Beta 1)

```

struct AnalyzeTextIntent: AppIntent {
    static var title: LocalizedStringResource = "Text analysieren"

    @Parameter(title: "Eingabe")
    var inputText: String

    func perform() async throws -> some IntentResult & ProvidesDialog {
        let session = LanguageModelSession()
        var fullResponse = ""

        for try await partial in session.streamResponse(to: "Analysiere: \ (inputText)") {
            fullResponse = partial.content
        }

        return .result(dialog: IntentDialog(stringLiteral: fullResponse))
    }
}

```

Per-Intent Privacy Declarations (neu in iOS 27) ✅ Bestätigt

```

struct HealthAnalysisIntent: AppIntent {
    static var title: LocalizedStringResource = "Gesundheitsdaten analysieren"

    // Pflicht für Enterprise- und Healthcare-Apps
    static var aiPrivacyPolicy: AIPrivacyPolicy = .onDeviceOnly

    func perform() async throws -> some IntentResult {

```

```

// Wird ausschließlich On-Device ausgeführt — kein Cloud-Routing
let session = LanguageModelSession()
// ...
return .result()
}
}

```

#### Migration SiriKit → App Intents

```

// — Vorher: SiriKit (deprecated)
class OrderIntentHandler: NSObject, INSendMessageIntentHandling {
    func handle(intent: INSendMessageIntent,
                completion: @escaping (INSendMessageIntentResponse) -> Void) {
        completion(INSendMessageIntentResponse(code: .success, userActivity: nil))
    }
}

```

#### // — Nachher: App Intents

```

struct SendMessageIntent: AppIntent {
    static var title: LocalizedStringResource = "Nachricht senden"

    @Parameter(title: "Empfänger")
    var recipient: ContactEntity

    @Parameter(title: "Nachricht")
    var message: String

    func perform() async throws -> some IntentResult {
        try await MessageService.send(message, to: recipient.id)
        return .result()
    }
}

```

---

### Siri Extensions API Experimentell (Beta 1)

Mit der **Siri Extensions API** kann eine App als alternativer KI-Anbieter in Siri registriert werden — gleichwertig mit Google Gemini, Anthropic Claude oder OpenAI ChatGPT.

**Zielgruppe:** Apps mit eigenem KI-Backend (KI-Assistenten, LLM-Wrapper, Enterprise-KI).

⚠️ API ist in Beta 1 experimentell. Breaking Changes sind wahrscheinlich.

#### import SiriExtensions

```

class MyAIExtension: SiriExtension {

    func handle(request: SiriExtensionRequest) async throws -> SiriExtensionResponse {
        let response = try await MyAIBackend.query(request.userQuery)
    }
}

```

```

    return SiriExtensionResponse(
        content: response.text,
        suggestedFollowUps: response.suggestions
    )
}
}

```

#### Info.plist Eintrag:

```

<key>NSExtensionAttributes</key>
<dict>
    <key>NSExtensionPointIdentifier</key>
    <string>com.apple.siri-extension</string>
    <key>SiriExtensionDisplayName</key>
    <string>Mein KI-Dienst</string>
</dict>

```

Nutzer wählen den Dienst unter **Einstellungen** → **Siri & Suchen** → **KI-Erweiterungen**.

---

### Writing Tools API Stabil (seit iOS 18, erweitert in iOS 27)

Writing Tools erscheinen **automatisch** in UITextView, WKWebView und NSTextView — kein Code erforderlich für die Basis-Integration.

#### Opt-out / Opt-in

```

textView.writingToolsBehavior = .none // Deaktiviert
textView.writingToolsBehavior = .limited // Eingeschränkt (kein Inline-Rewrite)
textView.writingToolsBehavior = .complete // Standard — vollständig aktiviert

```

#### Custom Context via Delegate

```
textView.writingToolsDelegate = self
```

#### **extension** MyViewController: UIWritingToolsDelegate {

```

    func writingToolsWillBegin(_ writingTools: UIWritingTools) {
        // UI vorbereiten
    }

    func writingTools(
        _ writingTools: UIWritingTools,
        requestContextFor range: UITextRange
    ) -> UIWritingToolsContext {
        return UIWritingToolsContext(
            attributedString: textView.attributedText,
            selectedRange: range
        )
    }
}

```

Neu in iOS 27: Grammatikprüfung (Proofreading) — automatisch in allen unterstützten Textfeldern.

---

## Image Playground API Stabil (seit iOS 18, erweitert in iOS 27)

*SwiftUI-Sheet*

```
import ImagePlayground
import SwiftUI
```

```
struct ContentView: View {
    @State private var showImagePlayground = false
    @State private var generatedImage: UIImage?

    var body: some View {
        VStack {
            if let image = generatedImage {
                Image(uiImage: image).resizable().aspectRatio(contentMode: .fit)
            }
            Button("Bild generieren") { showImagePlayground = true }
        }
        .imagePlaygroundSheet(isPresented: $showImagePlayground) { image in
            generatedImage = image
        }
    }
}
```

*Programmatische Generierung  Experimentell (Beta 1)*

```
guard ImageGenerator.isAvailable else { return }
```

```
let generator = ImageGenerator()
let image = try await generator.generate(
    prompt: "Ein futuristischer Stadtpark bei Sonnenuntergang",
    style: .illustration,
    aspectRatio: .square
)
```

Neu in iOS 27: Realistische Bildstile, nahezu beliebige Bildgrößen, ChatGPT-Integration für zusätzliche Stile.

---

## Verfügbarkeit & Geräteanforderungen

Framework

Min. iOS

Min. Chip

Beispielgerät

Einschränkung

Foundation Models (Text)

27.0

A17 Pro

iPhone 15 Pro

Ältere Chips: reduzierte Kapazität

Foundation Models (Multimodal)

27.0

A17 Pro

iPhone 15 Pro

iPhone 16 Pro empfohlen

Core AI Framework

27.0

A13

iPhone 11

Alle iOS-27-Geräte

App Intents (iOS-27-Features)

27.0

Alle

Alle iOS-27-Geräte

-

Siri Extensions API

27.0

Alle

Alle iOS-27-Geräte

Experimentell

Writing Tools

18.0

Alle

Alle iOS-18-Geräte

iOS 27: Grammatik ergänzt

Image Playground

18.0

A17 Pro

iPhone 15 Pro

iOS 27: neue Stile

Apple Intelligence gesamt

27.0

A17 Pro

iPhone 15 Pro+

-

**A13 (iPhone 11, SE 2. Gen):** iOS 27 unterstützt — Core AI Framework und App Intents verfügbar.

Foundation Models: eingeschränkt oder nicht verfügbar.

**A14–A16 (iPhone 12–15):** Core AI + App Intents verfügbar. Foundation Models eingeschränkt.

**A17 Pro+ (iPhone 15 Pro+):** Alle Features vollständig verfügbar.

**Graceful Degradation — immer implementieren:**

```
func performAIFeature() async {  
    if LanguageModelSession.isAvailable {  
        await useFoundationModels()  
    } else {  
        await showBasicFallback()  
    }  
}
```

---

## MLX - Open-Source ML Framework Bestätigt (Platforms State of the Union)

Quelle: WWDC 2026 Platforms State of the Union (09. Juni 2026)

**MLX** ist Apples Open-Source-ML-Framework für Entwickler und Researcher. In iOS/macOS 27 gibt es folgende Neuerungen:

Feature

Details

### **Metal 4 Support**

MLX nutzt Metal 4 für beschleunigtes Training

### **Multi-Mac-Training**

Modelltraining über mehrere Macs via RDMA over Thunderbolt

### **Apple Silicon optimiert**

Speziell für M-Chips und Neural Engine ausgelegt

**Relevanz für Entwickler:** - LLM-Experimente und Custom-Modell-Training direkt auf Mac (lokal, kein Cloud-Compute nötig) - Gemeinsames Training über mehrere Mac-Stationen (für Forschung und Enterprise) - GitHub: [github.com/ml-explore/mlx](https://github.com/ml-explore/mlx) und [github.com/ml-explore/mlx-swift](https://github.com/ml-explore/mlx-swift)

---

## Breaking Changes & Migrationspfad

 Dieses Kapitel wird bei jeder Beta aktualisiert. Geänderte APIs werden mit  **BREAKING CHANGE** markiert.

**Status aller KI-  
Frameworks (Beta 1 - 08.  
Juni 2026)**

Framework

Status

Stabilität

Bemerkung

Foundation Models Framework

 Beta

Breaking Changes möglich

LanguageModelSession, @Generable neu


Core AI Framework

 Beta

Kernarchitektur stabil

MCP-Connector experimentell

App Intents (Basis)

 Stabil seit iOS 16

Stabil

iOS-27-Erweiterungen experimentell

App Intents Streaming

 Beta

Breaking Changes möglich

Neu in iOS 27


Per-Intent Privacy Declarations

 Beta

Breaking Changes möglich

Neu in iOS 27


Siri Extensions API

 Experimentell

Breaking Changes wahrscheinlich

Frühe Beta


Writing Tools API

 Stabil seit iOS 18

Stabil

iOS 27: Grammatik ergänzt

Image Playground API

 Stabil seit iOS 18

Stabil

iOS 27: neue Stile

SiriKit

 Deprecated

Funktioniert noch

2-3 Jahre Migrationsfenster

Core ML

 Deprecated

Funktioniert noch

2-3 Jahre Migrationsfenster

## **Prioritätsliste für Entwickler**

Priorität

Aufgabe

Begründung

 Kritisch

UIScene-Lifecycle adoptieren

App startet ohne Adoption nicht (TN3187)

 Kritisch

SiriKit → App Intents

SiriKit deprecated, neue Siri-Features nicht erreichbar

 Hoch

Foundation Models evaluieren

Jetzt APIs kennenlernen, Beta nutzen

 Hoch

Core AI Framework in neuen Projekten

Core ML deprecated

 Hoch

Privacy Manifest: Siri-Routing deklarieren

Pflicht für Enterprise- und Healthcare-Apps

 Mittel

writingToolsBehavior prüfen

Automatisch aktiv — ggf. in spez. Textfeldern deaktivieren

 Mittel

Image Playground API evaluieren

Stabil seit iOS 18

Abwarten

Siri Extensions API

Beta 1 noch experimentell

Abwarten

Foundation Models Tool Calling in Produktion

Beta-API — erst nach Stabilisierung einsetzen

## Game Porting Toolkit Bestätigt (Platforms State of the Union)

Quelle: WWDC 2026 Platforms State of the Union (09. Juni 2026)

Das **Game Porting Toolkit** erhält in WWDC 2026 bedeutende Verbesserungen:

Feature

Details

### KI-Coding-Agents

Automatisierte Game-Porting-Workflows via Xcode-Agents

### Metal-Kommandozeilen-Tools

Neue Metal-CLI-Tools für Build-Pipelines

### Metal 4

Neue Graphics-API-Generation mit verbesserter GPU-Nutzung

**Relevanz für Entwickler:** - Windows-Spiele schneller auf macOS portieren - Metal 4 bietet niedrigere CPU-Overhead und bessere GPU-Auslastung - KI-Agents übernehmen repetitive Porting-Aufgaben


---

## App Store & TestFlight & AppStore Connect

### App Store SDK-Fristen

Datum

Pflicht

28. April 2026 

iOS 26 SDK (Xcode 26) – bereits in Kraft

~April 2027

iOS 27 SDK (Xcode 27) – erwartet

Empfehlung: Umstieg auf iOS-27-SDK direkt nach Stable-Release (September 2026).

### DeclaredAgeRange Framework Bestätigt

Neues datenschutzschützendes Altersverifikations-API (gesetzliche Pflicht in Texas/Australien):

**import DeclaredAgeRange**

**let** ageRange = **try await** DeclaredAgeRange.request()

### App Store Connect API 4.3.1 Bestätigt

**Veröffentlicht: 12. Mai 2026**

Aktuelle API-Version. CI/CD-Pipelines auf diese Version aktualisieren.

### TestFlight 4.2.1 Bestätigt

**Datum:** 29. Mai 2026

Aktuelle stabile Version. Accessibility-Verbesserungen (VoiceOver, Voice Control, Larger Text).

### App Store Connect 3.2 Bestätigt

Accessibility-Verbesserungen für VoiceOver und Voice Control

Unterstützung für 11 neue Sprachen beim App-Bewerben (Bangla, Gujarati, Kannada, Malayalam, Marathi, Odia, Punjabi, Slovenian, Tamil, Telugu, Urdu)

Verbesserte TestFlight-Feedback-Filterung

### Siri Extensions App Store Bereich Bestätigt

Für iOS 27: Dedizierter App-Store-Bereich für Siri-Extensions-kompatible Apps geplant. Preisgestaltung für KI-gestützte Interaktionen noch nicht angekündigt.

### Neue Richtlinien Bestätigt

Ab 4. Juni 2026: Neue Apple Accounts in Texas unterliegen Altersverifizierungspflichten für App-Downloads und In-App Purchases

Rückerstattungen ab 1. Januar 2026: In Euro, unabhängig vom Kaufdatum

---

## Bekannte Probleme

Bekannte Probleme werden nach Veröffentlichung der offiziellen Beta-1-Release-Notes auf [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/) ergänzt.

**Erwartete Problembereiche (basierend auf Beta-1-Erfahrungen früherer Versionen):** - Liquid-Glass-Rendering in gemischten UIKit/SwiftUI-Projekten - Custom Navigation Bar/Tab Bar Layouts - safeAreaInsets nach Floating-Tab-Bar-Einführung

**Bekannte Einschränkungen Beta 1 (basierend auf experimentellen Markierungen):** - @Generable (Foundation Models): 🚧 Experimentell — Breaking Changes möglich - Tool Calling (Foundation Models): 🚧 Experimentell — Breaking Changes möglich - App Intents Streaming: 🚧 Experimentell — Breaking Changes möglich - Siri Extensions API: 🚧 Experimentell — Breaking Changes wahrscheinlich - MCP-Connector (Core AI): 🚧 Experimentell — Breaking Changes möglich - Programmatische Image Generation: 🚧 Experimentell — Breaking Changes möglich

**Empfehlung:** Offizielle Release Notes auf [developer.apple.com/documentation/ios-ipados-release-notes/ios-ipados-27-release-notes](https://developer.apple.com/documentation/ios-ipados-release-notes/ios-ipados-27-release-notes) nach Beta-Download prüfen.

---

## Migration auf Apple 27

### Priorisierte Migrations-Checkliste

*Kritisch (App startet nicht ohne Adoption)*

**UIScene-Lifecycle adoptieren** ✅ Bestätigt (TN3187)

SceneDelegate erstellen

UIApplicationSceneManifest in Info.plist ergänzen

UIWindow(windowScene:) verwenden

*Hoch (UI bricht oder verhält sich falsch)*

**Liquid Glass Migration abschließen** ✅ Bestätigt

UIDesignRequiresCompatibility = true entfernen

Custom Backgrounds in UINavigationController, UITabBar, UIToolbar prüfen

safeAreaInsets nach Floating-Tab-Bar-Einführung neu evaluieren

Tab-Bar-Suche: Eigene Implementierungen gegen neue Tab-Bar-Integration testen

*Mittel (Deprecated APIs)*

**SceneKit → RealityKit** ✅ Bestätigt

**UIScreen.main ersetzen** ✅ Bestätigt

**actionSheet → confirmationDialog** ✅ Bestätigt

**TLS-Kompatibilität prüfen** ✅ Bestätigt

*Neu in iOS 27 (sofort evaluieren)*

**SiriKit → App Intents migrieren** ✅ Deprecation bestätigt

Zeitfenster: 2–3 Jahre

Neue Siri-Funktionen nur via App Intents

**Core ML → Core AI Framework evaluieren** ✅ Deprecation bestätigt

Neue KI-Features nur via Core AI Framework

Bestehende Core-ML-Implementierungen funktionieren weiterhin

**Privacy Manifest: Siri-Routing deklarieren** ✅ Bestätigt

Insbesondere für Enterprise- und Healthcare-Apps Pflicht

**Adaptive Layout APIs implementieren** ✅ Bestätigt

UISplitViewController / NavigationSplitView vorbereiten

Hinge-State-Detection integrieren

*macOS-spezifisch*

**CI/CD auf Apple Silicon prüfen** ✅ Bestätigt

**Rosetta-2-Abhängigkeiten identifizieren** ✅ Bestätigt

macOS 27: Rosetta 2 noch vollständig verfügbar

macOS 28 (Herbst 2027): Rosetta 2 weitgehend eingestellt

Jetzt native Apple-Silicon-Builds vorbereiten

*Geräteunterstützung*

**Deployment-Target prüfen** — Mindest-Deployment-Target iOS 27 erfordert A13 Bionic (iPhone 11, SE 2. Gen und neuer)

A13-Geräte werden in iOS 27 weiterhin unterstützt (laut WWDC 2026 Keynote)

---

## Fazit

Die WWDC 2026 (8.-12. Juni 2026) unter dem Motto „**All Systems Glow**“ hat iOS 27, iPadOS 27, macOS 27 (Golden Gate), watchOS 27 und visionOS 27 offiziell angekündigt. Developer Beta 1 steht sofort zur Verfügung.

### Die wichtigsten Erkenntnisse für Entwickler:


**SiriKit ist deprecated** – App-Intents-Migration jetzt einplanen (2-3 Jahre Zeitfenster). ✅ Bestätigt


**Core ML wird durch Core AI Framework ersetzt** – neue KI-Features müssen über Core AI Framework implementiert werden. ✅ Bestätigt


**Siri Extensions API** – eigene Apps als Siri-KI-Provider positionieren (Claude, Gemini, ChatGPT). ✅ Bestätigt


**UIScene-Lifecycle** – ohne Adoption startet die App beim Build gegen iOS-27-SDK nicht mehr (TN3187). ✅ Bestätigt


**Liquid Glass vollständig umsetzen** – UIDesignRequiresCompatibility = true fällt in Xcode 27 weg. ✅ Bestätigt

**Adaptive Layout APIs + iPhone App Resizing** – jetzt für iPhone Fold und iPad-Resizing vorbereiten.   
Bestätigt

**macOS Golden Gate: Apple Silicon only** – Intel-Unterstützung endet, Rosetta 2 läuft letztmals vollständig.  
 Bestätigt


**Performance-Fokus** – iOS 27 ist ein „Snow Leopard“-Release: bis zu 80% schnellerer AirDrop, 30% schnellere App-Starts.  Bestätigt


**Privacy Manifest für Siri** – on-device vs. Cloud-Routing deklarieren.  Bestätigt


**Foundation Models: AFM 3 (20B Sparse), Multimodal, LanguageModel-Protokoll, Built-In Tools** – Anbieter ohne Code-Änderungen wechseln, BarcodeReader, OCR, Spotlight-RAG.  Bestätigt

**Foundation Models Open Source** – Apple plant Open-Source-Release für Sommer 2026.  Bestätigt

**Private Cloud Compute kostenlos** – für Entwickler mit <2M Erstdownloads.  Bestätigt

**Swift 6.4: OS-Kernel in Swift, suppressible Warnings** – anyAppleOS, async defer, URL 4× schneller.    
Bestätigt

**SwiftUI: Toolbar APIs, Reorderable Containers, Document API, Spatial Preview Framework** – neue stabile APIs sofort einsetzbar.  Bestätigt

**Xcode 27: 30% kleiner, Apple-Silicon-exklusiv, iCloud Sync, Device Hub, On-Device AI Completion** – Privacy-first Code Completion direkt auf dem Mac.  Bestätigt

**MLX: Metal 4 + Multi-Mac-Training via RDMA** – LLM-Training direkt auf Mac-Infrastruktur.  Bestätigt

**Alle Build-Nummern Beta 1 (08. Juni 2026) – bestätigt via [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/):**

Plattform

Build

iOS 27 Beta 1

24A5355q

iPadOS 27 Beta 1

24A5355q

macOS Golden Gate Beta 1

26A5353q

watchOS 27 Beta 1

24R5289n

tvOS 27 Beta 1

24J5289o

visionOS 27 Beta 1

24M5291p

Xcode 27 Beta 1

27A5194q

,

*Letzte Aktualisierung: 09. Juni 2026 – WWDC 2026 Platforms State of the Union (AFM 3, LanguageModel-Protokoll, Built-In-Tools, MLX Metal 4, Xcode 30% kleiner, Swift OS-Kernel, Core AI Python-Tools, Spatial*

Preview, iPhone App Resizing, Game Porting Toolkit ergänzt)

Quellen: WWDC 2026 Keynote (Apple), WWDC 2026 Platforms State of the Union (09. Juni 2026), [developer.apple.com/news/releases/](https://developer.apple.com/news/releases/), [developer.apple.com/swift/whats-new/](https://developer.apple.com/swift/whats-new/), [developer.apple.com/swiftui/whats-new/](https://developer.apple.com/swiftui/whats-new/), [developer.apple.com/apple-intelligence/whats-new/](https://developer.apple.com/apple-intelligence/whats-new/), [developer.apple.com/xcode/whats-new/](https://developer.apple.com/xcode/whats-new/), MacRumors, 9to5Mac, TechTimes ([techtimes.com/articles/318039](https://techtimes.com/articles/318039), [318045](https://techtimes.com/articles/318045)), [ofox.ai](https://ofox.ai), [byteiota.com](https://byteiota.com), Apple Accessibility Announcement (19. Mai 2026), Apple Newsroom ([apple.com/newsroom](https://apple.com/newsroom)), BusinessWire