



*Conoscete
Windows
Bridge?*

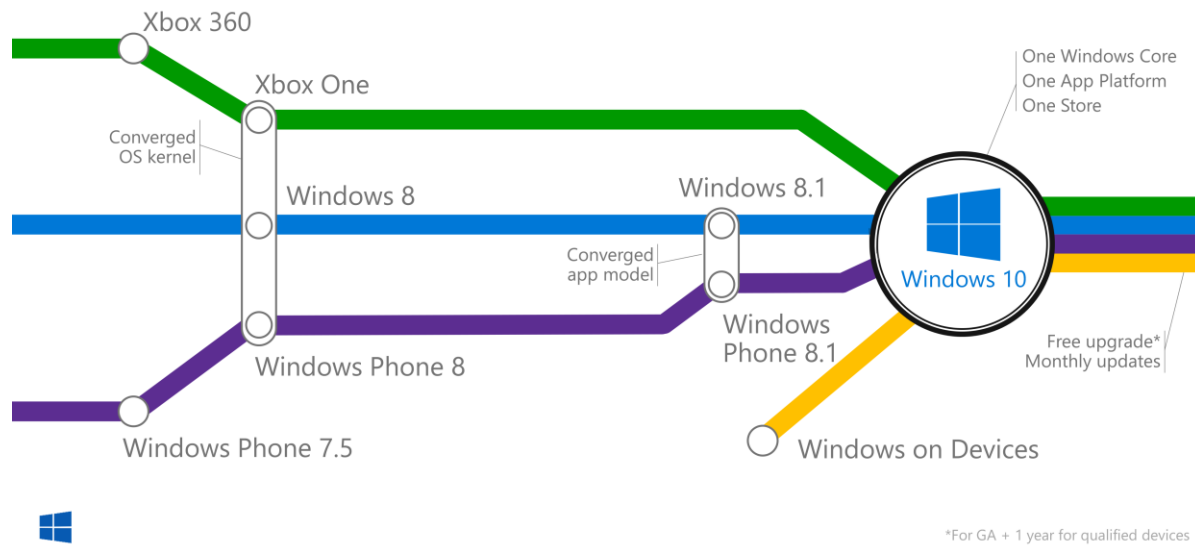
PATERNO' GAETANO



Mail: tanopaterno@gmail.com

Facebook: gaetano.paterno.77

A bigger opportunity with one Windows



Con Windows 8, Microsoft ha introdotto Windows Runtime (WinRT), un'evoluzione dell'app model di Windows, che doveva essere un'architettura applicativa comune, ma non lo è stato del tutto. Al rilascio di Windows Phone 8.1 è stato effettuato l'allineamento di Windows Runtime tra Windows Phone 8.1 e Windows. Gli sviluppatori hanno quindi potuto creare *app universali di Windows 8* destinate sia a Windows che a Windows Phone con una base di codice condivisa.



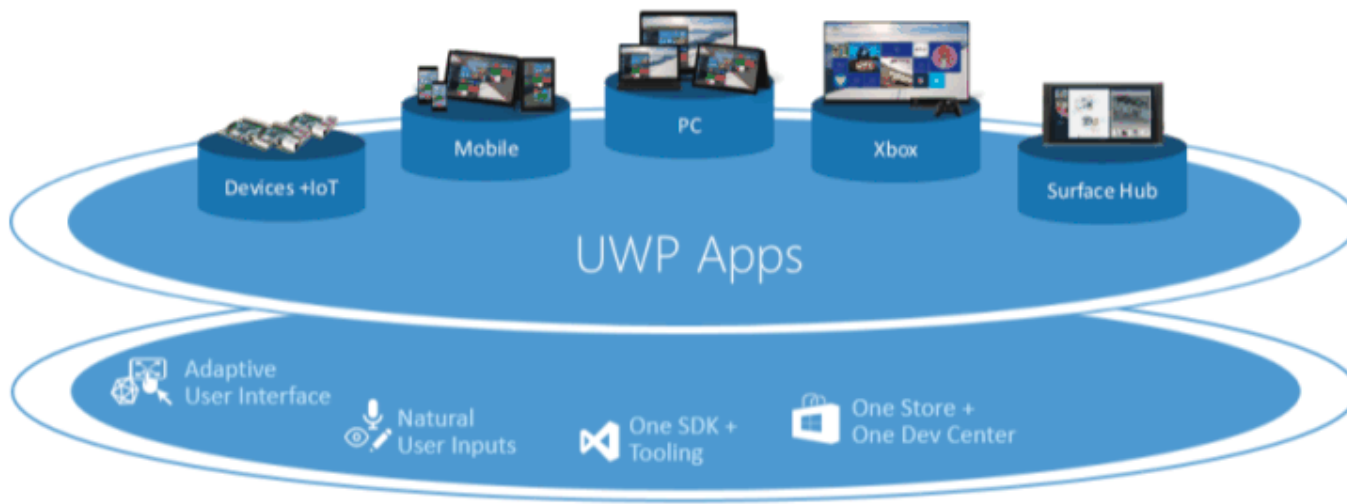
PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



One Windows Platform



Windows 10 introduce la piattaforma UWP (Universal Windows Platform), che evolve ulteriormente il modello di Windows Runtime e lo introduce nella memoria centrale unificata di Windows 10. Essendo parte della memoria centrale di Windows 10, UWP offre ora una piattaforma per app comune, disponibile su tutti i dispositivi che lo eseguono. Grazie a questa evoluzione, le app destinate alla piattaforma UWP possono chiamare non solo le API WinRT comuni per tutti i dispositivi, ma anche le API specifiche, incluse le API Win32 e .NET, della famiglia di dispositivi in cui viene eseguita l'app.



One platform
One store
One experience

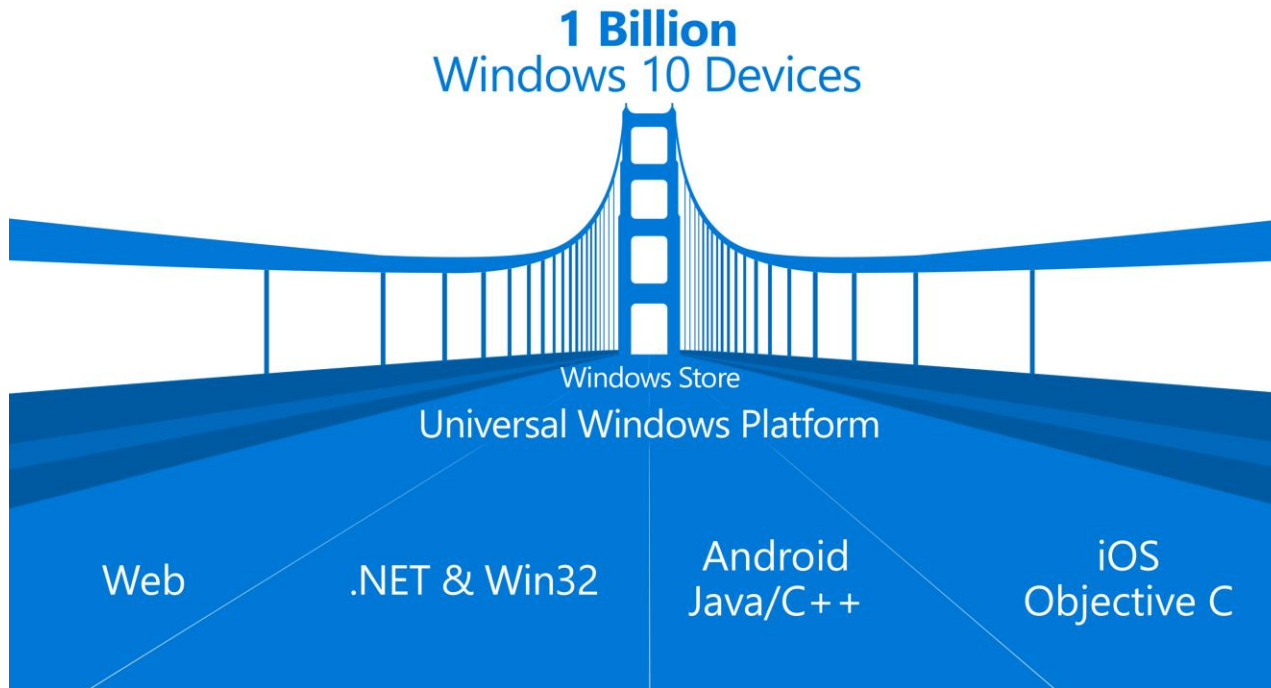


La piattaforma UWP offre un livello di API di base garantito su tutti i dispositivi. Questo ti permette di creare un singolo pacchetto dell'app che può essere installato su un'ampia gamma di dispositivi.

I controlli dell'interfaccia utente adattivi e i nuovi pannelli di layout introdotti in UWP, aiutano a personalizzare l'interfaccia utente per un'ampia gamma di risoluzioni dello schermo.

Inoltre, con un singolo pacchetto dell'app, Windows Store offre un canale di distribuzione unificato per raggiungere ogni tipo di dispositivo su cui può essere eseguita l'app.





**Porta il tuo codice in
Windows
indipendentemente dalla
destinazione per cui era
pensato il tuo codice.**



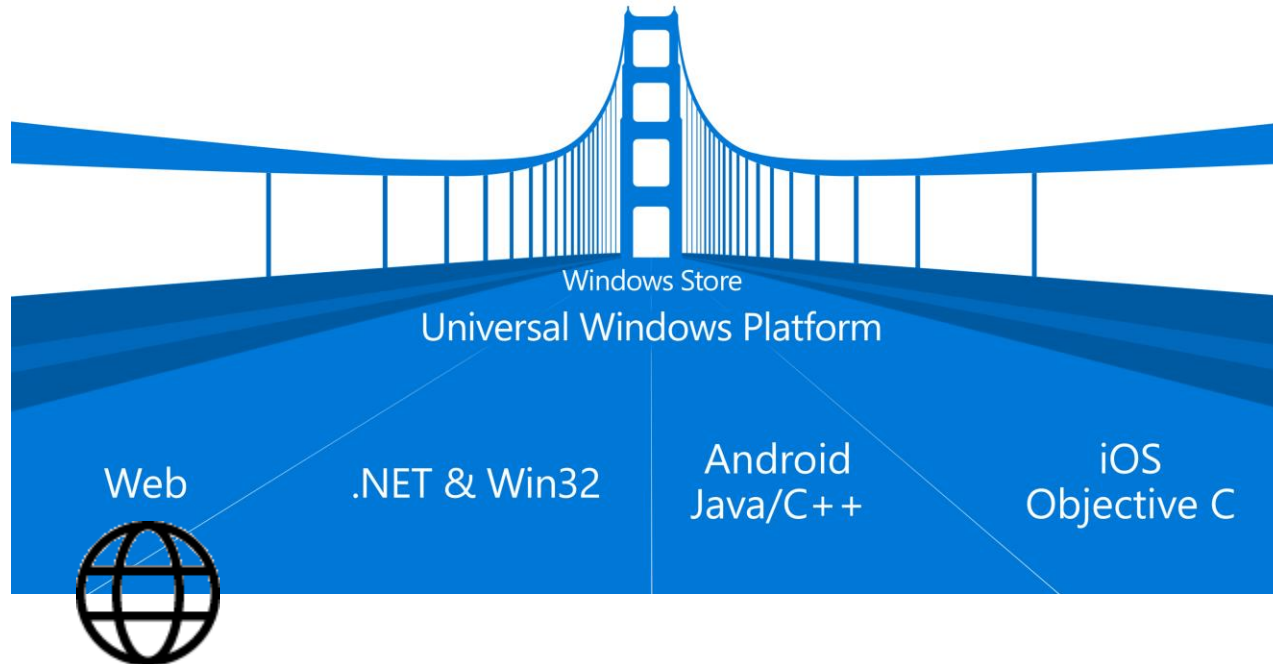
PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



1 Billion
Windows 10 Devices



WEB

Per i siti Web o le app Web, è possibile usare Visual Studio 2015 per creare facilmente un'app di Windows 10 che raccoglie in un pacchetto il tuo sito Web o la tua app Web per la pubblicazione in Windows Store. Questo tipo di app, nota come Hosted Web App, può essere usata per chiamare le API UWP direttamente dal codice JavaScript del tuo sito Web e creare un'esperienza utente più coinvolgente.



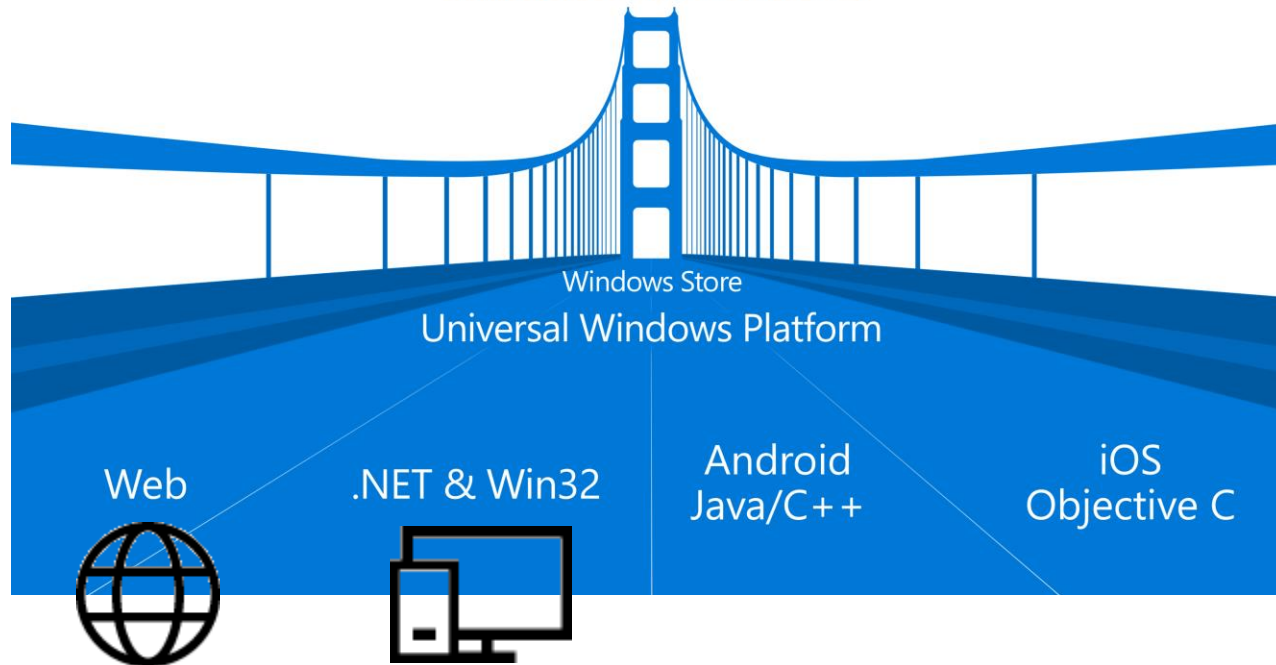
PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



1 Billion
Windows 10 Devices



WINDOWS

Se hai un'app in Silverlight per Windows Phone, è possibile usare il [bridge Silverlight di Mobilize.Net](#) per trasferirla sulla piattaforma UWP. Questo bridge è attualmente disponibile come anteprima per gli sviluppatori e offre un subset iniziale dei mapping di piattaforma per Silverlight e UWP.

Se hai un'applicazione .NET o basata su Win32, "Project Centennial" renderà possibili la creazione di un pacchetto per l'applicazione e la pubblicazione in Windows Store.

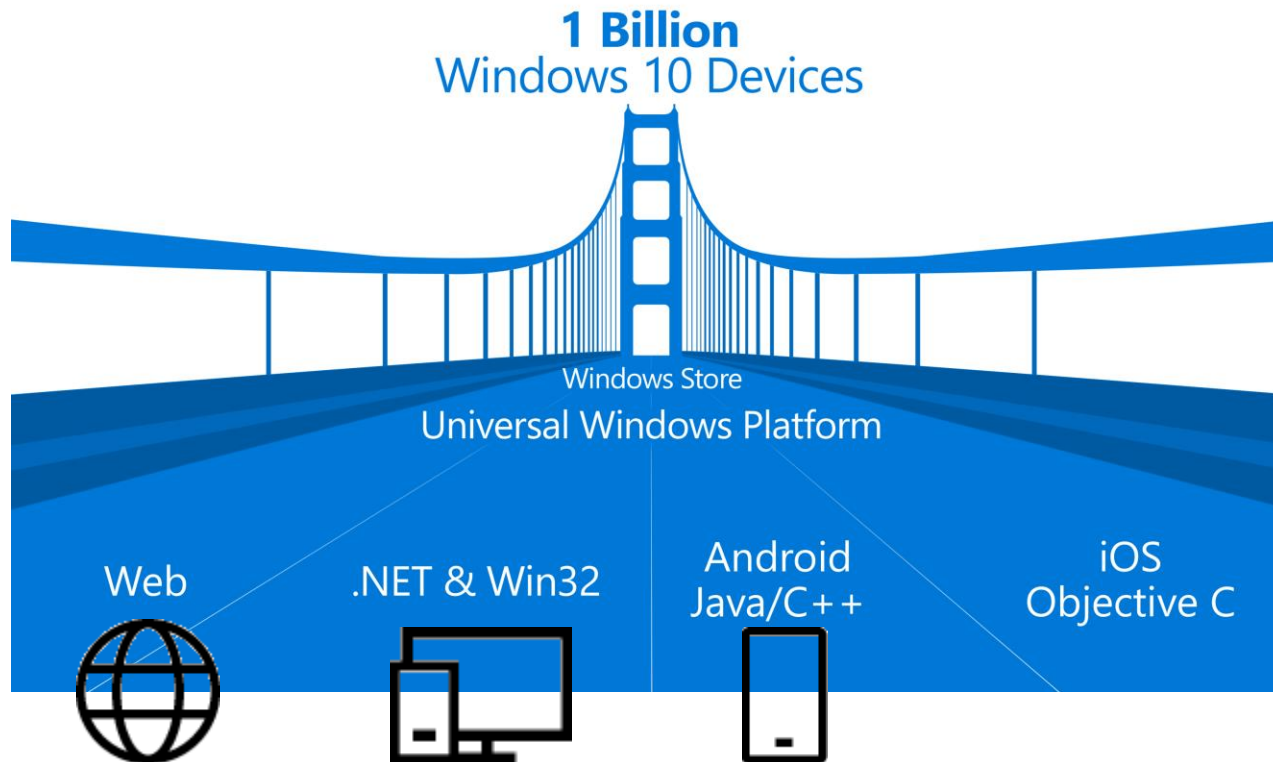


PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)





ANDROID

Se hai un'app per Android, puoi usare il bridge Android per renderla disponibile per i dispositivi Windows 10 Mobile usando le estensioni per Android SDK e vari ambienti IDE noti per Android. Per la maggior parte delle app sono necessarie poche modifiche al codice, se non nessuna.



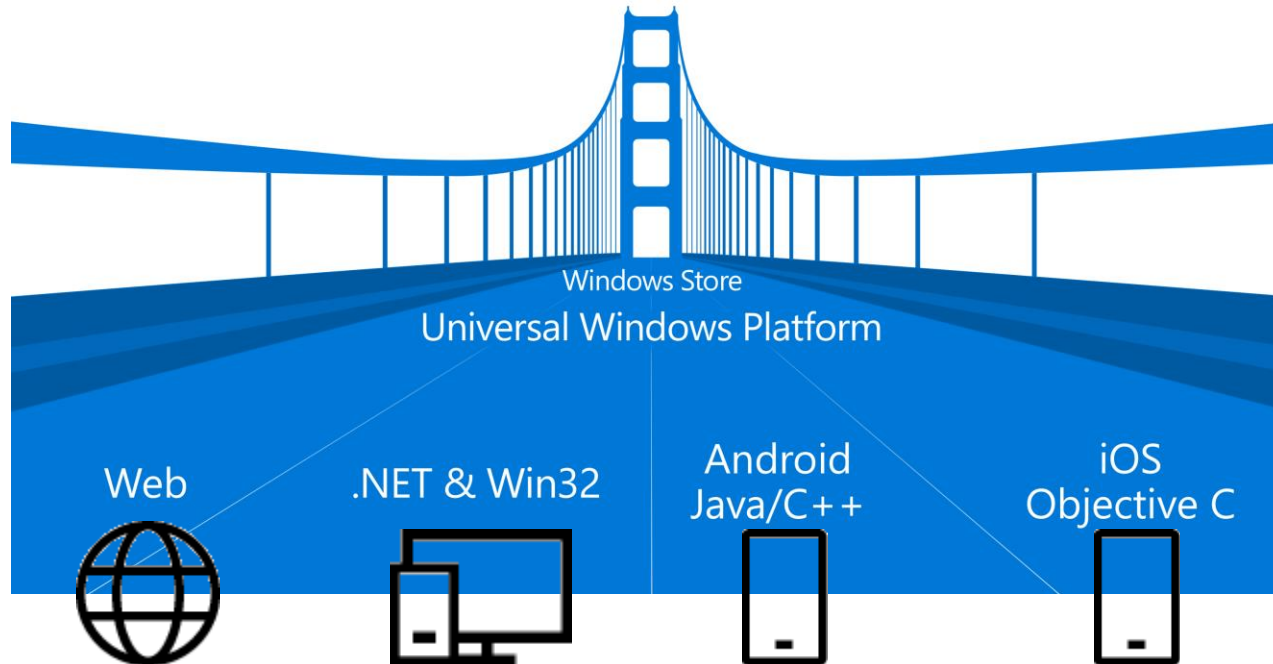
PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



1 Billion
Windows 10 Devices



iOS

Se hai un'app per iOS, puoi usare il bridge iOS per creare una versione UWP della tua app con Visual Studio 2015 e il codice Objective-C esistente. Puoi estendere il codice esistente con le funzionalità native di Windows, in base alle tue esigenze.



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



Build great
Windows apps
from your
Objective-C code

Windows 10



In questa sessione andremo a vedere un po' più approfonditamente ciò che è

Windows Bridge for iOS

(precedentemente denominato 'Progetto Islandwood'), come consente agli sviluppatori iOS di portare le loro competenze e il codice Objective-C per Windows e perché Microsoft ha deciso di dare questo particolare bridge di Windows disponibile come [open-source su GitHub](#).



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)





Windows Bridge for iOS e' stato rilasciato su GitHub in concomitanza con l'evento Build 2015 nel mese di aprile, con rilascio finale prevista per questo autunno.



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



Windows Bridge for iOS è concepito per consentire agli sviluppatori iOS di creare applicazioni Windows utilizzando il codice esistente in Objective-C e le sue competenze. A tale scopo, questo bridge è costituito da quattro componenti:

Objective-C compiler
un compilatore incluso in Visual Studio che prende il codice Objective-C e l'ho compila in una Windows Universal App nativa.

Objective-C runtime
per fornire funzionalità avanzate al linguaggio.

iOS API headers/libs
sulla base delle API di Objective-C, vengono fornite delle API che danno un'ampia compatibilità con le API iOS.

Visual Studio IDE Integration
infine viene fornito un tool che importa il progetto Xcode in Visual Studio 2015.





Why the bridge is not a port

L'obiettivo del bridge per iOS non è mai stato semplicemente quello di eseguire applicazioni iOS su Windows.

Piuttosto l'obiettivo è aiutare a scrivere grandi applicazioni *Windows* che utilizzano non molto di più del vostro codice già esistente e tutta la conoscenza dello sviluppatore.

Naturalmente, stanno continuando a lavorare per ampliare la compatibilità con iOS.



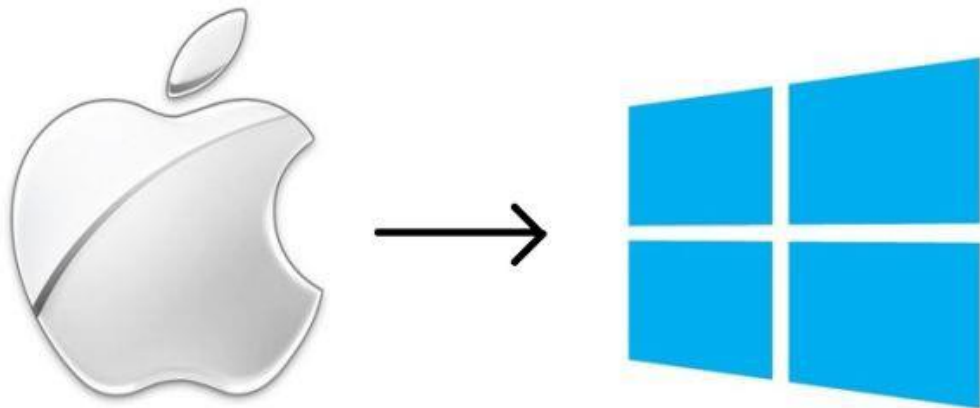
Dietro questo obiettivo ci sono tre principi fondamentali che hanno guidato l'architettura e il design del bridge per iOS:

Accesso completo alle Windows API
rende facile utilizzo delle Windows API all'interno di codice Objective-C.

compatibilità iOS
permettete agli sviluppatori di riutilizzare al massimo il codice esistente fin dove possibile.

Nessun sandboxing
iOS e le API di Windows devono essere in grado di lavorare insieme.



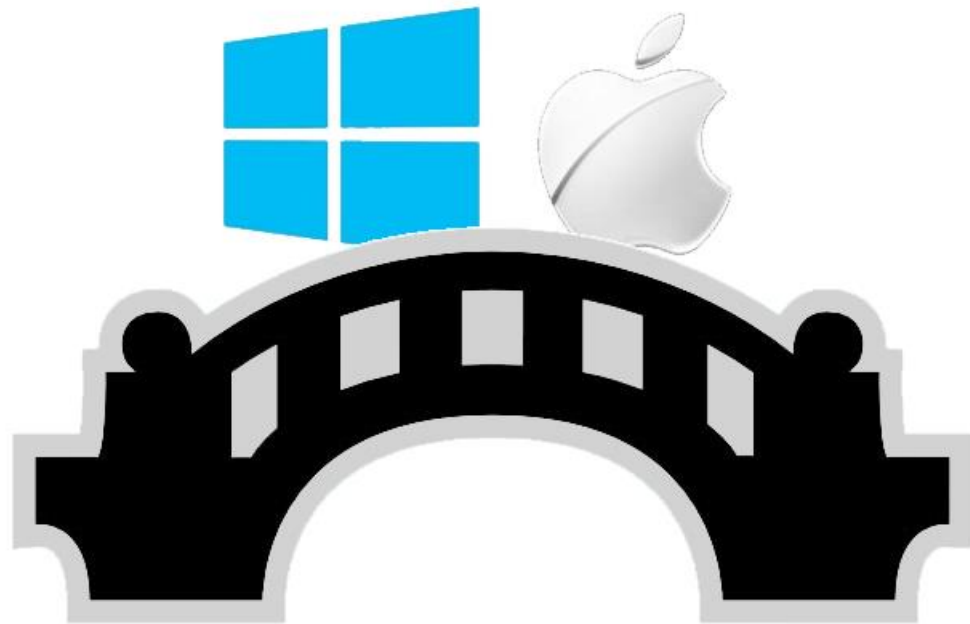


Il primo e il terzo principio sono collegati e particolarmente importanti.

Il primo principio perché Windows ha un set di API completamente funzionale e ricco, che continua a crescere ed evolversi.

Il terzo perché avere la possibilità di chiamare e utilizzare interamente le API di Windows sarebbe severamente limitata se il set di API sarebbero in modalità sandbox uno da altro, limitando la capacità di compilare buone applicazioni Windows in Objective-C.





Quando le API Windows e Objective-C si incontrano

Il bridge utilizza un compilatore personalizzato (clang + cl) per compilare codice sorgente Objective-C, e i file oggetti generati sono poi collegati insieme utilizzando il linker di Microsoft.

Questo approccio è ottimo perché permette a Objective-C e C++/CX di poter coesistere nello stesso progetto e chiamarsi tra loro utilizzando le interfacce di C o C++.

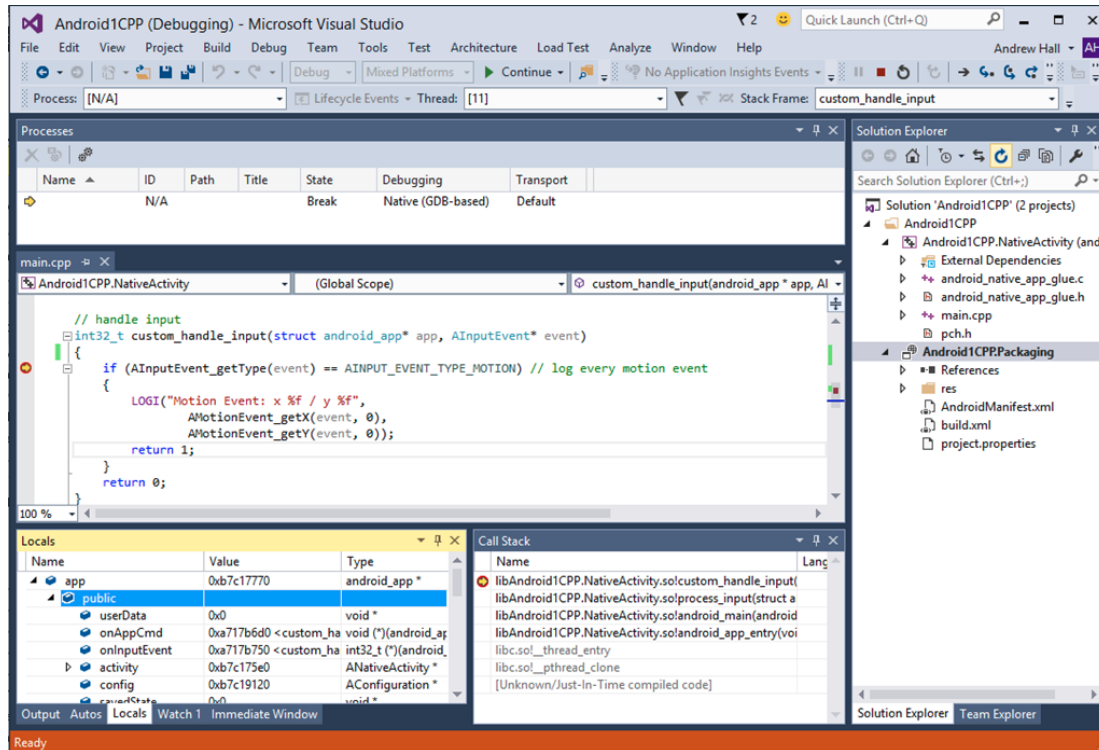


PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



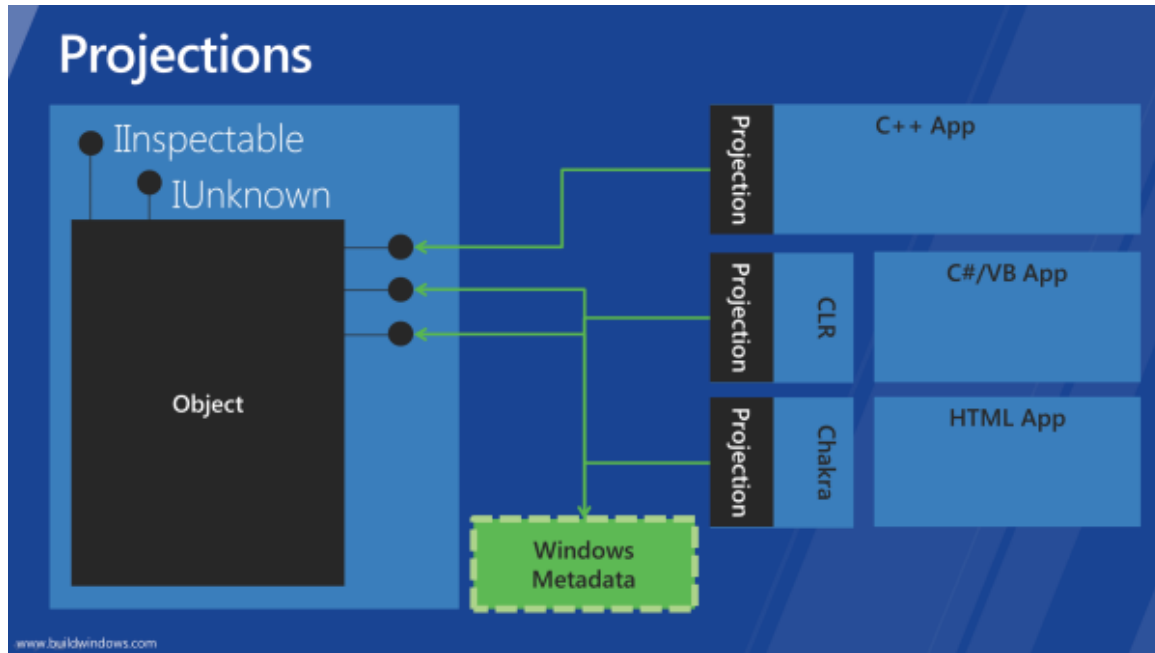


L'approccio funziona, ma c'è una piccola complicazione con esso.

Perché clang non "capisce" le estensioni CX (che sono necessari per chiamare le API di Windows), quindi è sarebbe necessario creare il file cpp e quindi connettere manualmente Objective-C e C++/ CX in modo che si possa sfruttare appieno l'insieme delle API Windows Universal Platform (UWP).

Mentre questo è perfettamente fattibile, in Microsoft hanno creduto che avrebbero potuto fare di meglio e questo è stato possibile con le "proiezioni" (projections).





Per chi è nuovo, in breve, ciò che noi chiamiamo "bindings" Microsoft ora lo chiama "projections". Le proiezioni sono il processo di esposizione delle API per tre ambienti: nativo (C e C++), HTML/JavaScript e .NET. Se si crea un componente in C++ o un linguaggio .NET, le sue API verranno memorizzate in un file WinMD e saremo in grado di consumarle da tutti e tre gli ambienti (nativi, JavaScript e .NET). Per il bridge hanno preso e usato questo stesso schema per consentire di utilizzare le API di Windows direttamente da Objective-C. Attualmente il bridge consente di utilizzare la gran parte delle API UWP e si spera di raggiungere l'intero utilizzo.



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



Ad esempio, esaminiamo come è possibile richiamare in modo asincrono un browser nella vs app utilizzando il bridge:

Opzione 1 (invocando il browser dal codice c++/cx):

```
1 auto uri = ref new  
2 Windows::Foundation::Uri\("http://www.example.com"\);  
3 concurrency::task<bool> launchUriOperation  
4 (Windows::System::Launcher::LaunchUriAsync(uri));  
5 launchUriOperation.then([](bool success)  
6 {  
7     if (success)  
8     {  
9         // URI launched  
10    }  
11    else  
12    {  
13        // URI launch failed  
14    }  
15 });
```

Opzione 2 (invocando la stessa API dal codice Objective-C):

```
[WSLauncher launchUriAsync:[WUri  
createUri:@"http://www.example.com/"] success:nil failure:nil];
```





XAML e UIKit: insieme all'ultimo

Ora che è possibile chiamare le API di Windows da Objective-C, per non limitare l'utilizzo di tale set di API, invece di implementare un compositor separato per gli elementi iOS/UIKit, l'intera applicazione utilizza il compositor XAML, con CALayers (che sono alla base essenzialmente di ogni vista in iOS) legato agli elementi XAML corrispondenti.



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

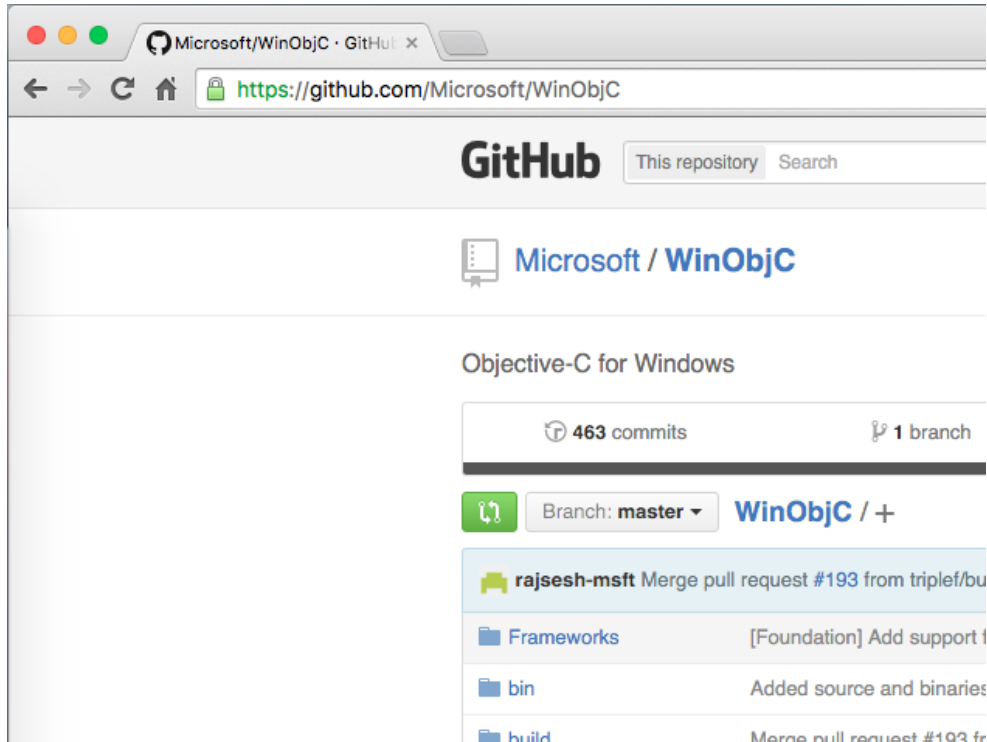
Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)

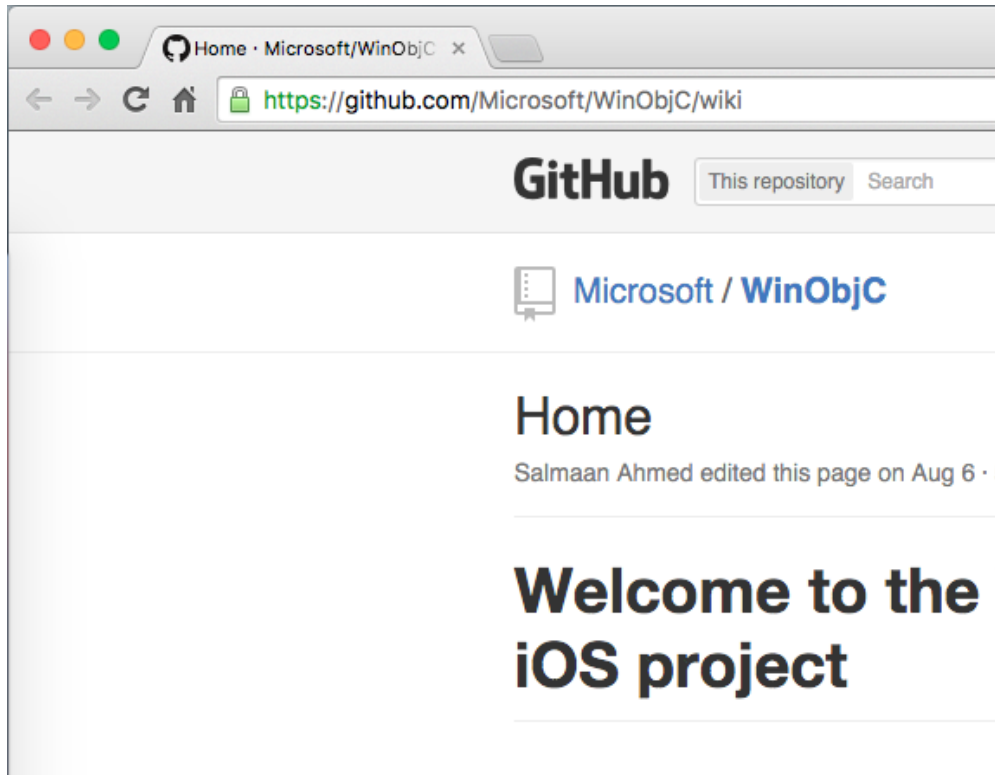


Perchè Windows Bridge for iOS è disponibile su GitHub

Microsoft ha messo a disposizione su GitHub tutti i sorgenti del framework e dei tools per permettere agli sviluppatori di contribuire in tanti modi al progetto WinObjC, come?

- Segnalando un bug
- Verificando la correzione dei bug
- Inviando codice per la correzione di un bug
- Inviando una richiesta di funzionalità
- Inviando una prova di funzionalità
- Dire agli altri sul progetto WinObjC
- Dire gli sviluppatori quanto apprezzi il progetto





Wiki

Su GitHub è disponibile anche una pagina Wiki che guida passo-passo su come scaricare, installare e usare l'sdk ed il tool **vsimporter**.

Il tool **vsimporter** consente di importare il progetto Xcode in una nuova soluzione UWP, con supporto ad Objective-C, per Windows 10.

Se si vuole una soluzione per Windows 8.1, utilizzare l'opzione `-format` e specificare uno dei target: `winstore8.1`, `winphone8.1` o `winstore10` (`winstore10` è il predefinito).

Per ulteriori informazioni utilizzare l'opzione `-help` nella riga di comando per vedere la serie completa di opzioni supportate.





DEMO



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



Che cosa è ancora in fase di sviluppo?

Dato che questo progetto è ancora in fase di sviluppo, ci sono alcune caratteristiche che non sono ancora disponibili:

- E' supportata solo la piattaforma x86, ARM è in arrivo
- Alcune ottimizzazioni/personalizzazioni del compilatore non funzionano correttamente, mandando in crash clang
- Il supporto storyboard (al momento l'interfaccia è da rifare)
- MapKit
- AssetsLibrary
- Rubrica
- Annunci
- Annotazioni Objective-C
- Cattura e riproduzione media

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)



PATERNO' GAETANO





Problemi o quesiti?

Potete raggiungere il team di sviluppo in vari modi:

- Tweeter @WindowsDev e segnare le domande con #winobjc
- Inviare domande sui post di StackOverflow con il tag winobjc
- Visitare il canale #winobjc su IRC (webchat.freenode.net)

Visitate i siti:

- <https://dev.windows.com/it-it/bridges>
- <https://dev.windows.com/it-it/bridges/ios>
- <https://github.com/Microsoft/WinObjC>
- <https://github.com/Microsoft/WinObjC/wiki>



PATERNO' GAETANO

Mail: tanopaterno@gmail.com

Facebook: [gaetano.paterno.77](https://www.facebook.com/gaetano.paterno.77)

