

# Microsoft Build



# Quoi de neuf pour les développeurs Windows ?

Christophe Lavallo – 30 juin 2020

# Christophe Laval

Développeur Senior



@ChristopheLav



<https://ca.linkedin.com/in/christophe-lavalle-817ba312b>




2019 ☺

# Au menu

- Command-Line
  - Windows Terminal 1.0, et au-delà
  - Windows Package Manager Preview (WinGet)
  - Windows Subsystem for Linux (WSL) 2, et au-delà
- Windows app platform
  - Project Reunion
  - WinUI 3
  - Quel avenir pour UWP ?
- Conclusion

# Windows Terminal 1.0, et au-delà

# Windows Terminal 1.0 finalisé

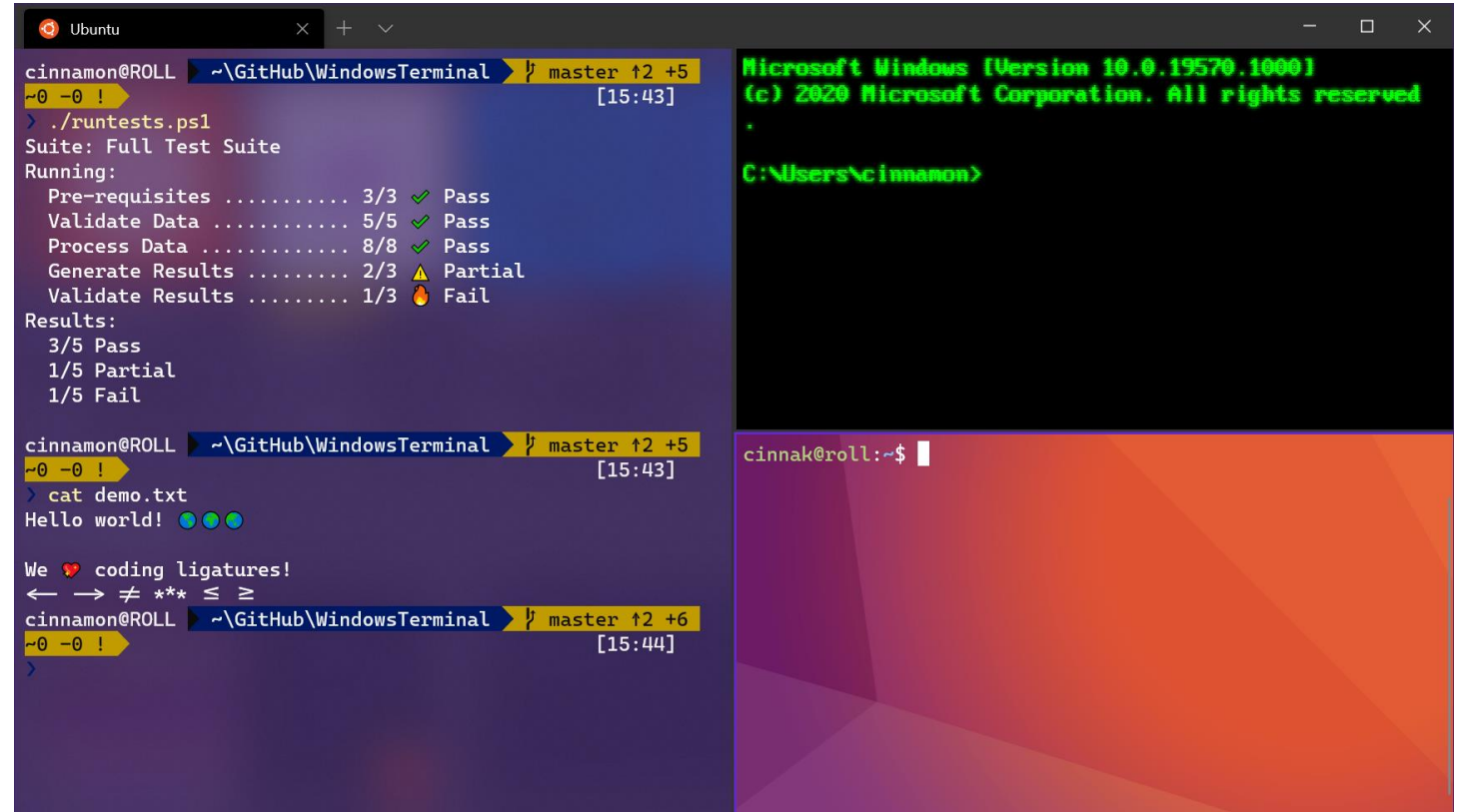
Un an après son annonce, la première version est finalisée 

Fonctionnalités :

- ▶ Rendu accéléré par le GPU
- ▶ Tabs et Panes
- ▶ Options de personnalisation

Requiert Windows 10 1903 (build 18362) ou plus récent

Disponibilité : Windows Store + MSIX





# Windows Terminal, ce qui s'en vient

- Windows Terminal
  - Versions stables
  - Mises à jour mensuelles
  - Première mise à jour en juillet 2020
  - [aka.ms/terminal](https://aka.ms/terminal)
- Windows Terminal (Preview)
  - Pour essayer les nouvelles fonctionnalités en cours de développement
  - Mises à jour mensuelles
  - Version 1.1 Preview disponible depuis juin 2020
  - [aka.ms/terminal-preview](https://aka.ms/terminal-preview)

# Démo

Windows Terminal 1.1 / Interface des paramètres



# Windows Terminal 2.0

Announce

- Mai/Juin 2021
- Reprise des fonctions reportées de la première version
- Ajout des demandes de la communauté
- Roadmap :
  - Interface graphique pour les réglages (le JSON persistera pour ceux qui apprécient 😊)
  - Mises à jour non destructives pour les paramètres
  - Personnalisation de l'interface (couleurs)
  - Command palette
  - Liens cliquables
  - Ouvrir en tant que « ... »
  - Et bien plus !

# Windows Package Manager Preview Aka WinGet

# Windows Package Manager (Preview)

Announce

*« Un outil command-line permettant aux développeurs de découvrir, installer, mettre à jour et configurer ses applications sur Windows 10 »*

```
> winget install vscode
```

# Démo

Windows Package Manager

# Windows Package Manager (Preview)

- Requiert Windows 10 1709, build 16299+
  - Inclus dans l'application native « App Installer » + disponible sur GitHub
- Supporte les installeurs EXE, MSI et MSIX
- Basé sur des manifests Yaml hébergés sur GitHub
  - <https://github.com/microsoft/winget-pkgs>

```
Id: Microsoft.VisualStudioCode
Version: 1.46.1.cd9ea64888
Name: Visual Studio Code
Publisher: Microsoft Corporation
Homepage: https://code.visualstudio.com
License: Copyright (c) Microsoft Corporation
Description: Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications.
AppMoniker: vscode
Commands: code
Installers:
  - Arch: x64
    Url: https://az764295.vo.msecnd.net/stable/cd9ea6488829f560dc949a8b2fb789f3cdc05f5d/VSCodeUserSetup-x64-1.46.1.exe
    InstallerType: inno
    Sha256: e29f73f5b6bad9ea5a484d9a934141a03d3fa39c55efd6e65e6e02dea6be9aa5
```

# Windows Package Manager, à venir

- Sources :

- Support du Windows Store comme source d'applications
- Support des ZIP comme « installeur »
- Support des repository exposés avec REST

- Utilisabilité

- Lister les applications installées
- Support des mises à jour des applications déjà installées
- Support de la désinstallation d'une application
- Configuration roaming entre les ordinateurs d'un même utilisateur
- Installation « system » ou « utilisateur »

**WSL 2, et au-delà**



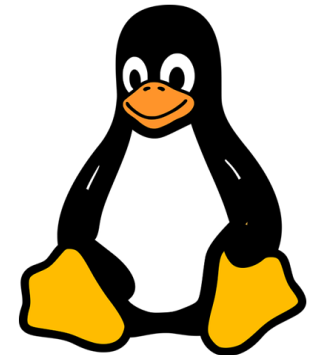
# WSL, retour d'expérience



# WSL 2 finalisé

- Disponible dans Windows 10, version 2004
- Un véritable kernel Linux dans Windows
- 100% des calls system supportés
  - On passe d'un système de traduction des appels vers Windows NT à un système basé sur la virtualisation (lightweight VM)
- Amélioration sensible de la performance des I/O
- Conversion des distributions WSL 1 vers WSL 2 possible

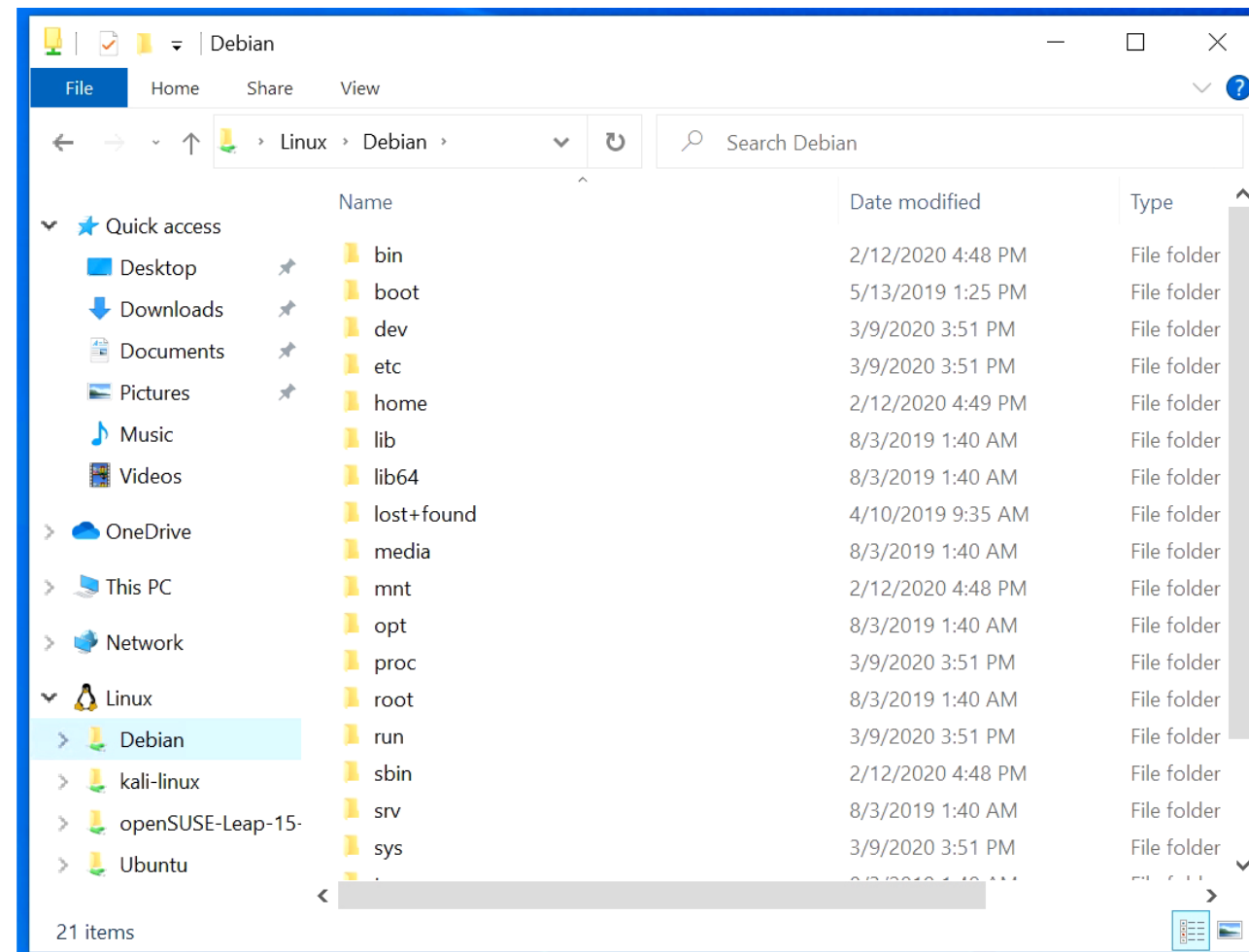
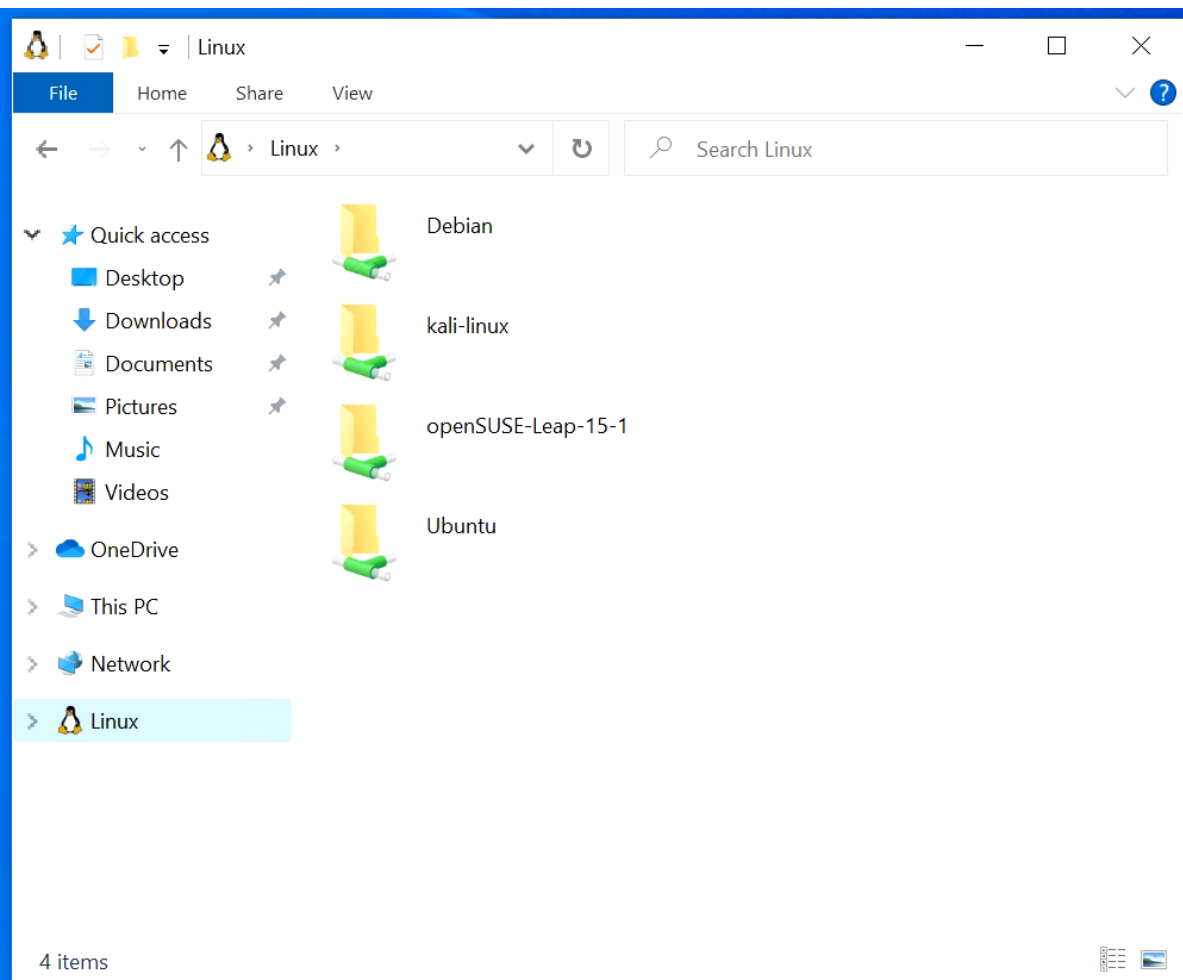
- ▶ Maintenance du kernel Linux via Windows Update
- ▶ Docker : WSL 2 remplace Hyper-V comme backend



# WSL, ce qui s'en vient

Announce

- Intégration dans l'explorateur de fichiers Windows



# WSL, ce qui s'en vient

Announce

Demande **#1** de la communauté depuis le lancement de WSL

▶ Support du GPU pour les calculs !

**CUDA**

Nvidia

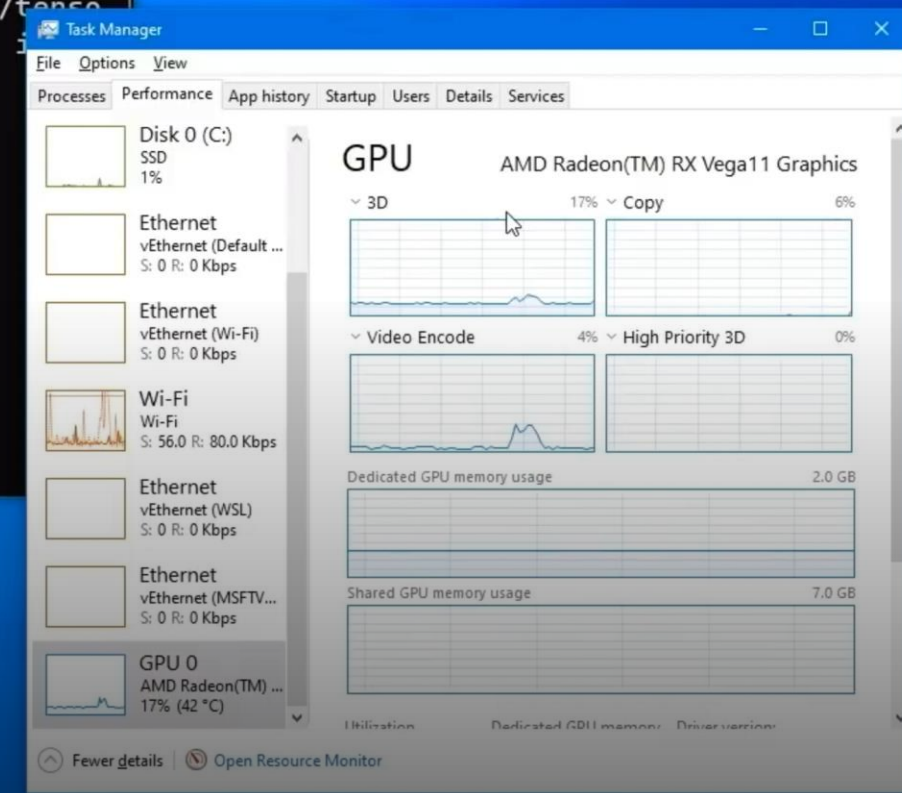
**DirectML**

Tous les GPUs DirectX 12

```
WARNING:tensorflow:From /home/test/miniconda3/envs/tfdml-1.15/lib/python3.6/site-packages/tensorflow_core/python/autograph/converters/directives.py:119: The name tf.parse_single_example is deprecated. Please use tf.io.parse_single_example instead.
```

```
WARNING:tensorflow:From /home/test/miniconda3/envs/tfdml-1.15/lib/python3.6/site-packages/tensorflow_core/python/autograph/converter/directives.py:119: The name tf.image.resize_images is deprecated. Please use tf.image.resize instead.
```

```
WARNING:tensorflow:  
The TensorFlow contrib module will not be included in TensorFlow 2.0.  
For more information, please see:  
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md  
  * https://github.com/tensorflow/addons  
  * https://github.com/tensorflow/io (for I/O related ops)  
If you depend on functionality not listed there, please file an issue.
```



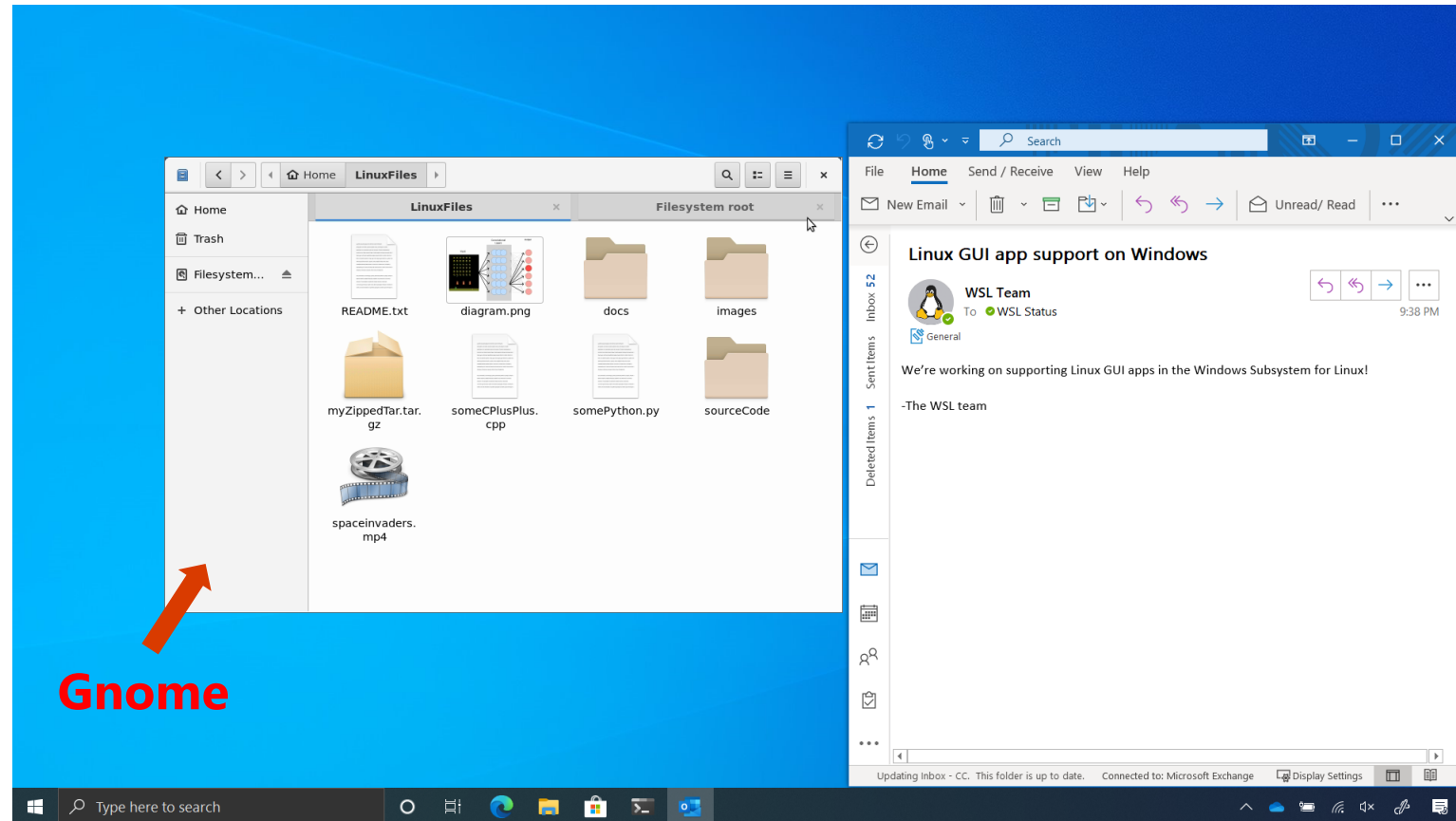
# WSL et GPU – Démo

# WSL, ce qui s'en vient

Announce

## ▶ Support natif des interfaces graphiques (GUI)

- Serveur Wayland intégré à WSL 2
- Communication via RDP pour l'accès aux applications depuis Windows
- Accélérées par le GPU
- DirectX : API Direct 3D 12 pour les rendus 3D disponible dans un environnement Linux



```
caloewen@DESKTOP-O70TFMS: x + v
wlslsld: Linux version: 4.19.119-microsoft-WSL2-standard+ (test@IOURIT-TEST2)
wlslsld: WLWSL system root: "/home/caloewen/.config/wlslsld/sysroot"
wlslsld: uid: 1000 euid: %d 1000 gid:1000 egid: 1000
wlslsld: failed rmdir(/tmp/.X11-unix)
wlslsld: spawning "/mnt/c/Windows/System32/cmd.exe" /c ver
wlslsld: Windows build: 10.0.19617.1000
wlslsld: Windows drive path: "/mnt/c"
wlslsld: cmd.exe at: "/mnt/c/Windows/System32/cmd.exe"
wlslsld: spawning "/mnt/c/Windows/System32/cmd.exe" /c set
wlslsld: Windows %APPDATA%: C:\Users\crloewen\AppData\Roaming
wlslsld: RDP client is rdclientwsl.exe
wlslsld: Installing rdp clients to Windows (if necessary)
wlslsld: host ip: 172.18.191.211
wlslsld: starting wlslsld...
wlslsld: Date: 2020-05-12 PDT
wlslsld: [17:33:56.644] xwayland-path is: /usr/bin/Xwayland
wlslsld: [17:33:56.648] using FreeRDP version 2.1.0
wlslsld: [17:33:56.762] TLS support activated
wlslsld: [17:33:56.767] Registered plugin API 'weston_rdp_outpu
wlslsld: [17:33:56.768] Registered plugin API 'weston_rdp_ext_a
wlslsld: [17:33:56.768] rdp_output_create rdp:0 0x563d8814bf50
wlslsld: [17:33:56.768] setting size for head rdp:0 640 480
wlslsld: [17:33:56.768] Output 'rdp:0' enabled with head(s) rdp
wlslsld: [17:33:56.768] rdprpc fd is 5
wlslsld: [17:33:56.768] Loading module '/home/vsts/local/lib/x8
wlslsld: [17:33:56.954] Registered plugin API 'weston_xwayland_
wlslsld: [17:33:56.954] Registered plugin API 'weston_xwayland_
wlslsld: [17:33:56.954] xserver listening on display :1
wlslsld: [17:33:56.954] waking wlslsld-compostor
```

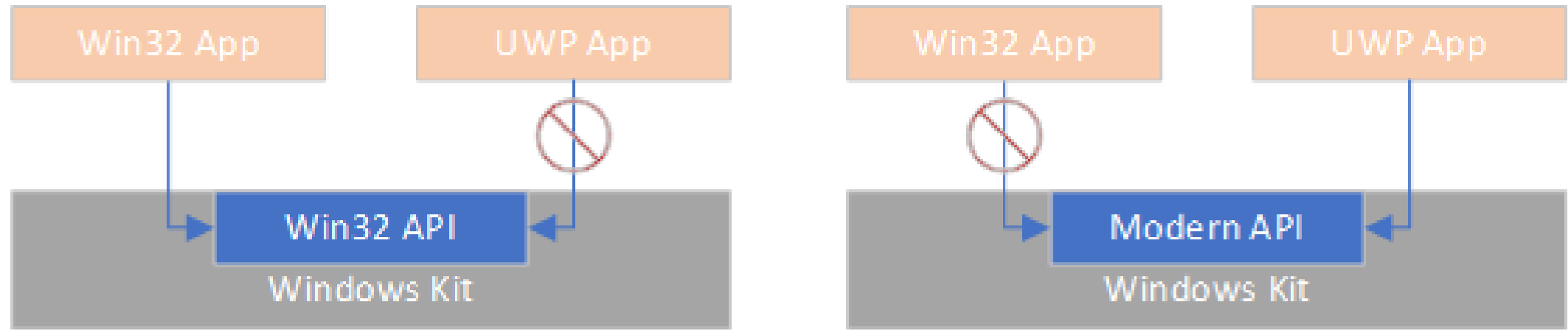
```
caloewen@DESKTOP-O70TFMS: x + v
caloewen@DESKTOP-O70TFMS:~/SpaceInvadersDeepQLearning/media$ cd ..
caloewen@DESKTOP-O70TFMS:~/SpaceInvadersDeepQLearning$ |
```

# WSL et GUI – Démo





« Project Reunion »

# Project Reunion, c'est quoi ?



# Project Reunion, composants

- Disponible aujourd'hui 
  - WinUI 3 for XAML
  - C++/WinRT, RUST/WinRT et C#/WinRT
  - MSIX-Core
- Disponible demain 
  - Edge/Chromium backed WebView2
  - Modern Lifecycle helpers
  - Startup Tasks
  - Update Scan Integration
  - Access to user resources
  - Modern Resource Tooling

**WinUI 3, et au-delà**

# État de l'art des Framework UI



**MFC**  
1992

**WinForms**  
2002

**WPF**  
2006

**UWP  
XAML**  
2012

**C++**

**.NET**

**C++ & .NET**

# Desktop (Win32) Apps



# UWP Apps



# Desktop (Win32) Apps



# UWP Apps



+

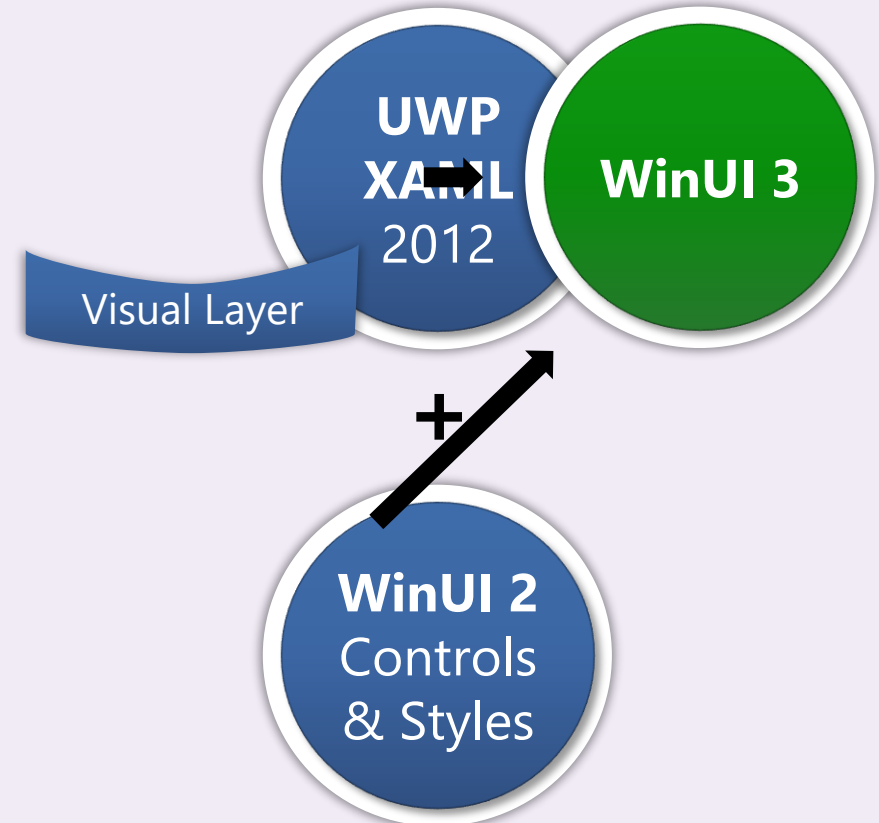




# Desktop (Win32) Apps



# UWP Apps



# Desktop (Win32) Apps



# UWP Apps



# Desktop (Win32) Apps

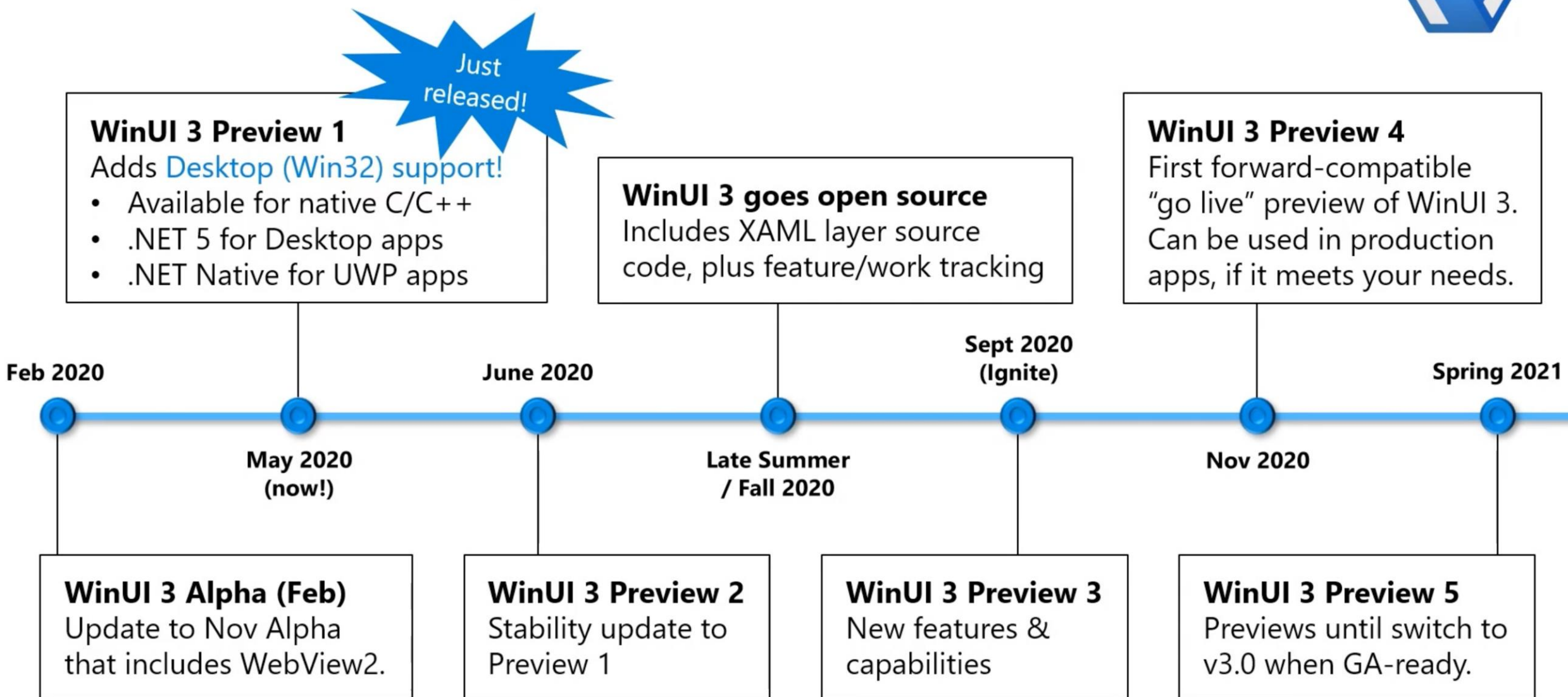
# UWP Apps



WinUI 3 sera un Framework UI moderne pour **chaque** développeur



# WinUI 3 Roadmap



# WinUI 3 Preview 1

- **Limitations et problèmes**

- Le contenu ne peut être hosté que dans une fenêtre par application
- XAML Islands absent
- Les outils de développement (Hot Reload, Live Tree, Designer)
- Des choses qui ne fonctionnent pas encore : ARM64, Xbox, HoloLens (cible : 2021)
- (...)

UWP – Quel avenir ?

# UWP, c'est quoi déjà ?

- Modèle applicatif moderne

- AppContainer / Sandbox
- Cycle de vie maîtrisé (suspension, redémarrage, etc.)
- Tâches en arrière-plan
- Notifications

- APIs système modernes

- Asynchrones
- Sécurisés / Permissions
- Accessibles en C++, C#, VB.NET et JavaScript

- XAML pour Framework UX

- Supporte les écrans Hi-DPI
- Adaptatif : les contrôles sont compatibles tant avec le clavier, la souris, le stylet ou encore les doigts

- Une distribution moderne

- Windows Store
- APPX / MSIX
- Installation et désinstallation propre

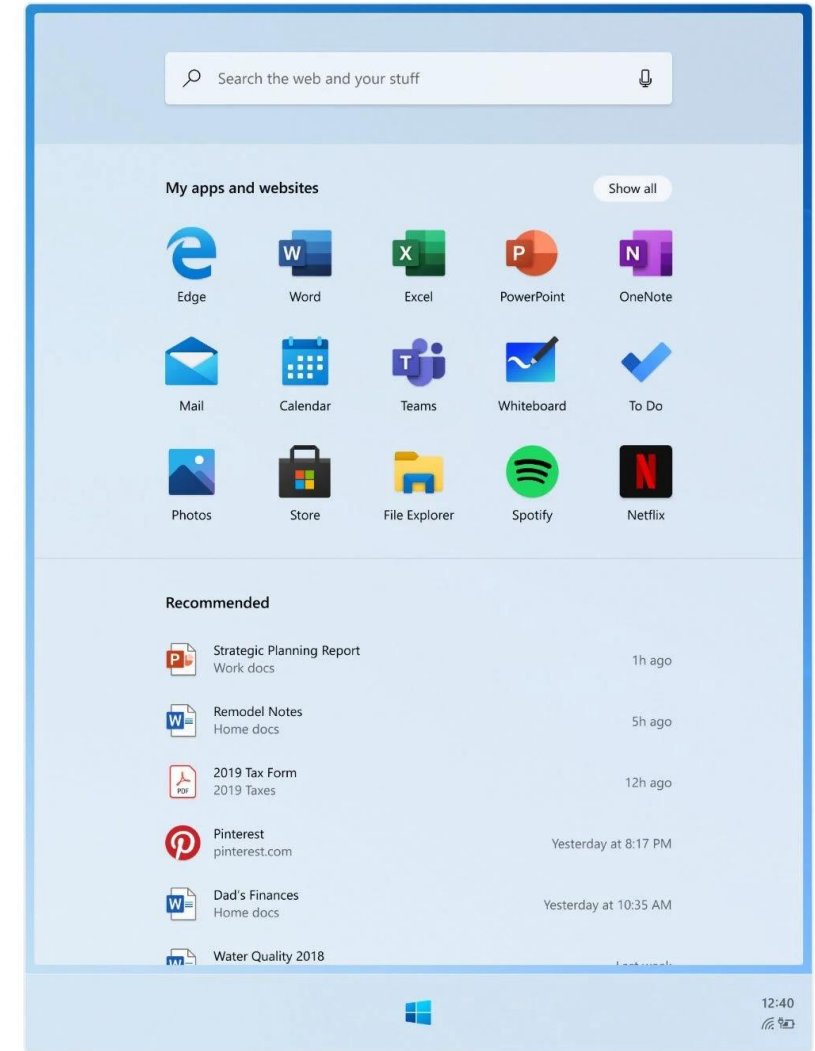
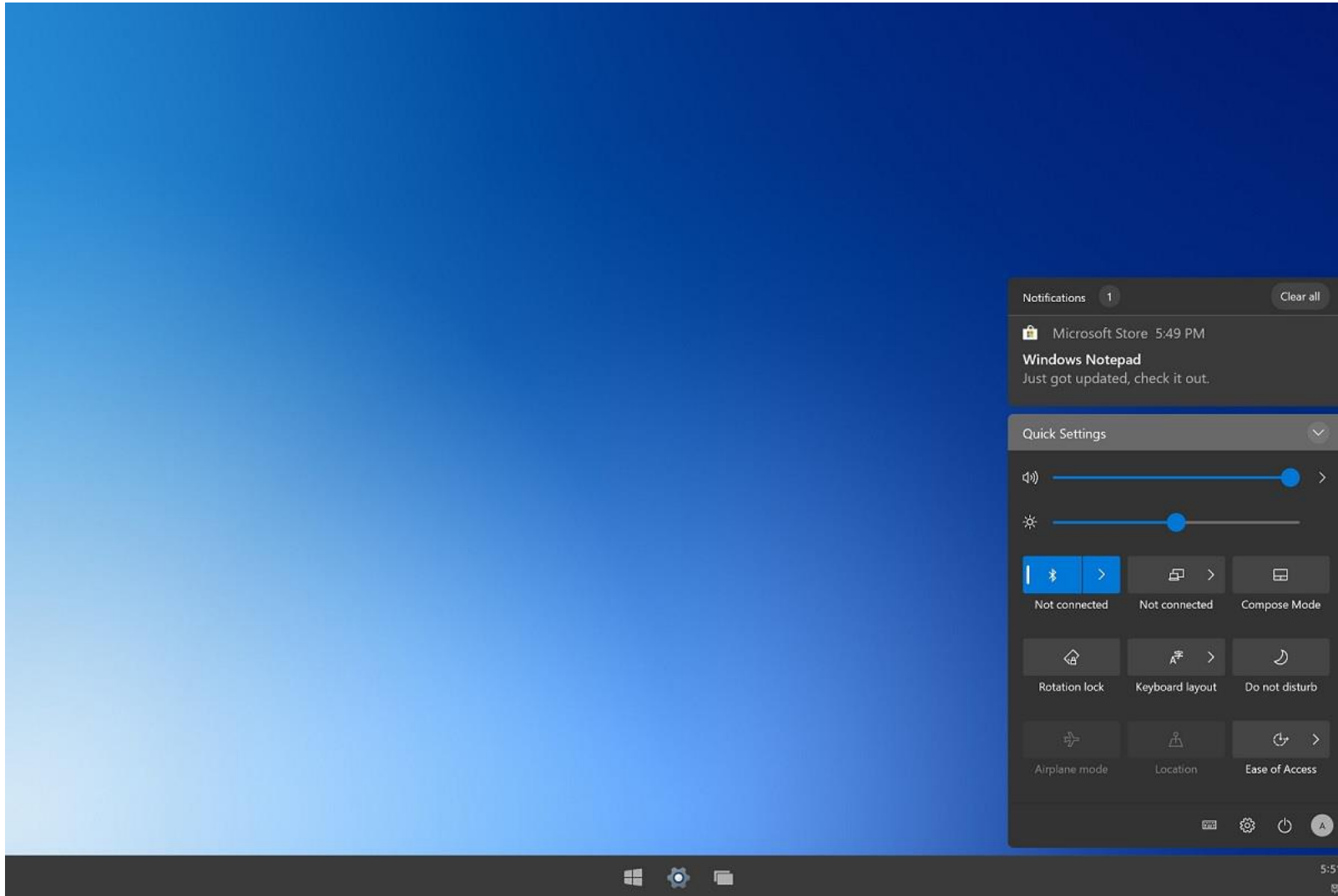


# UWP, quel avenir ?

- AppContainer est inclus dans Windows : il n'est pas déprécié et est la base de toutes les applications modernes
- UWP avec .NET Core : « c'est mort »
  - WinRT retiré de .NET Core (60k lignes de code)
  - .NET Native déprécié
  - UWP « .NET » n'évoluera plus, il n'avait déjà pas suivi les dernières évolutions

→ **WinUI viendra faciliter les ponts entre les applications .NET et « UWP »**
- UWP avec C++ : rien ne change, code natif utilisant directement les APIs Windows

# UWP, quel avenir ?      Windows 10X



# Conclusion

# Conclusion

- Faire de Windows le meilleur endroit pour les développeurs est un des focus principal de Microsoft
- La ligne de commande continue de redevenir « cool » : Terminal, WSL 2 et WinGet
- Windows app platform: Microsoft bascule vers une approche inclusive pour ces différentes bases techniques en unifiant leur accessibilité
- UWP n'est pas mort – même si le terme risque de s'effacer

Merci !

-

Questions ?



