Windows Phone
Security State of the Art?

Cyrill Bannwart

Compass Security Schweiz AG
Werkstrasse 20
Postfach 2038
CH-8645 Jona

Tel +41 55 214 41 60
Fax +41 55 214 41 61
team@csnc.ch
www.csnc.ch

Agenda

Introduction

The Windows View

- Windows Environment
- Attack Surface
- Breaking Out

The Mobile View

- Sandboxing
- Encryption & Enterprise Data Protection
- Windows Bridges

Conclusion

Introduction – Why Windows Mobile?

Third player, investing in it

Microsoft is a major player on the business desktop, servers and software

- Just missing the mobile part
- But attempting to catch up with the acquisition of Nokia, Xamarin, etc.
- Still understands / answers best companies’ needs

Global convergence

- Business & private
- Mobile & fix

Something new to look at (and maybe break? ;-)

- Our focus was the Windows Phone / Mobile platform itself

Introduction – Cyrill Bannwart

Joined Compass Security in 2013

- IT Security Analyst
- Security trainings teacher
- Mobile apps developer

Electrical Engineer with a strong interest in

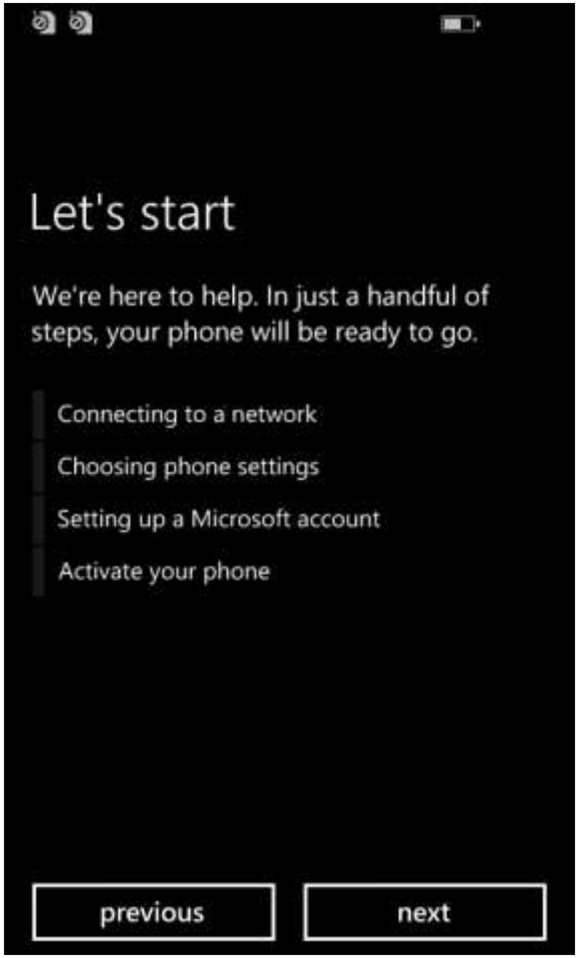
- Embedded devices
- Network communications

Mostly dealing with Network, Unix and (iOS) Mobile apps

Giving Security Trainings for

- Secure Mobile Apps (iOS & Android)
- iPhone & iPad Security

Let’s get started



Let's start

We're here to help. In just a handful of steps, your phone will be ready to go.

- Connecting to a network
- Choosing phone settings
- Setting up a Microsoft account
- Activate your phone

Agenda

Introduction

The Windows View

- **Windows Environment**
- **Attack Surface**
- **Breaking Out**

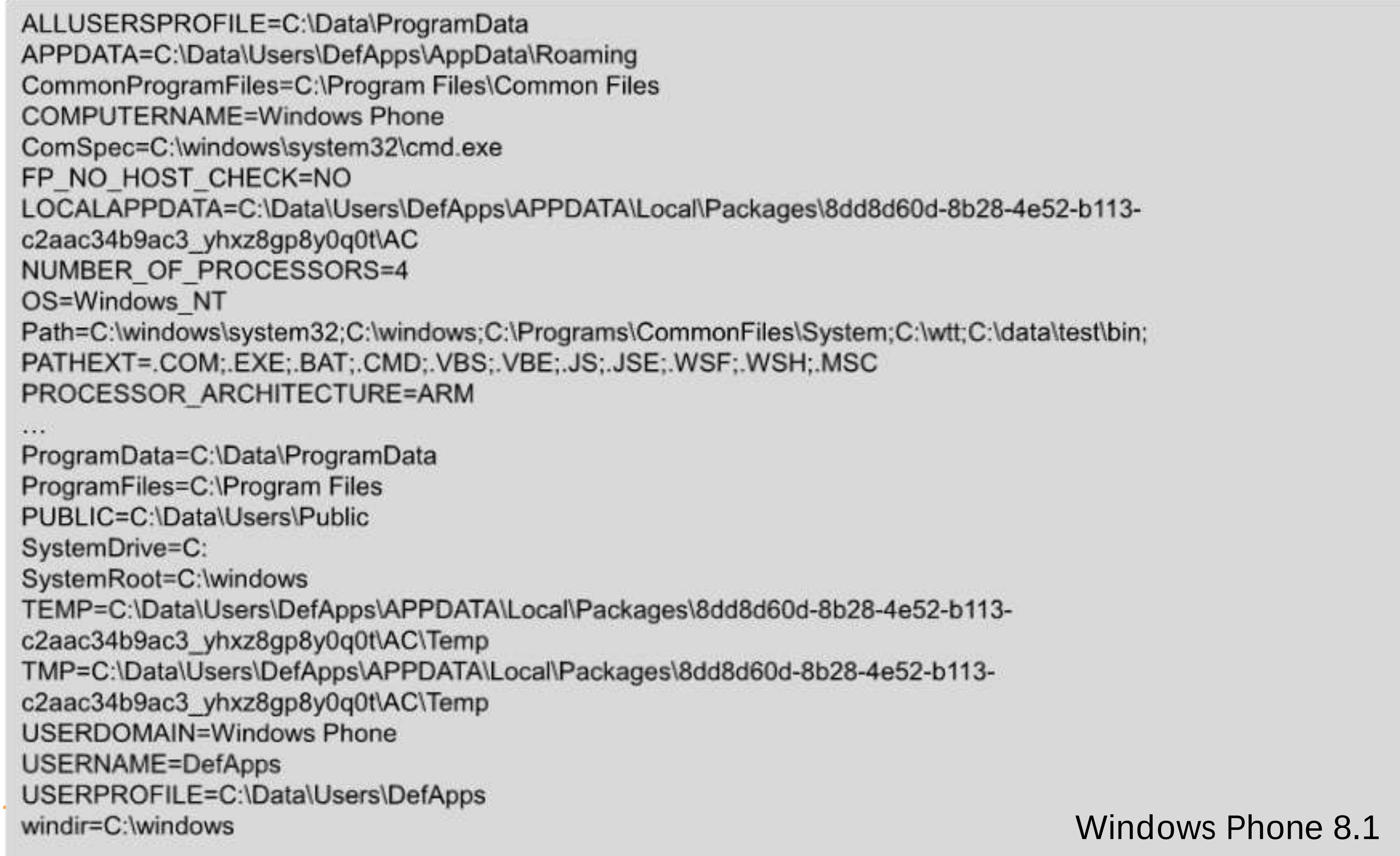
The Mobile View

- Sandboxing
- Encryption & Enterprise Data Protection
- Windows Bridges

Conclusion

The Windows View

Crash dumps are always useful and a good start…



```
ALLUSERSPROFILE=C:\Data\ProgramData
APPDATA=C:\Data\Users\DefApps\AppData\Roaming
CommonProgramFiles=C:\Program Files\Common Files
COMPUTERNAME=Windows Phone
ComSpec=C:\windows\system32\cmd.exe
FP_NO_HOST_CHECK=NO
LOCALAPPDATA=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC
NUMBER_OF_PROCESSORS=4
OS=Windows_NT
Path=C:\windows\system32;C:\windows;C:\Programs\CommonFiles\System;C:\wtt;C:\data\test\bin;
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE=ARM
…
ProgramData=C:\Data\ProgramData
ProgramFiles=C:\Program Files
PUBLIC=C:\Data\Users\Public
SystemDrive=C:
SystemRoot=C:\windows
TEMP=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC\Temp
TMP=C:\Data\Users\DefApps\APPDATA\Local\Packages\8dd8d60d-8b28-4e52-b113-
c2aac34b9ac3_yhxz8gp8y0q0t\AC\Temp
USERDOMAIN=Windows Phone
USERNAME=DefApps
USERPROFILE=C:\Data\Users\DefApps
windir=C:\windows
```

Windows Phone 8.1

Windows part – Overview

We focused on 3 aspects for the Windows part:

(Ab)Use of Windows Utilities and Features

- Can I gather information or perform undesired actions using built-in features?

Application Attack Surface

- Or how can I misuse Internet Explorer / Microsoft Edge to run unwanted code?

Development and APIs

- List the documented APIs and see what a developer might run as code

(Ab)Use of Windows Utilities and Features

A Windows desktop is user (and attacker) friendly

- Lots of information (eventlogs, detailed error messages, …)
- Lots of settings to influence (Control Panel, file & registry access, …)
- Built-in programs and features (notepad, sticky keys for accessibility, …)
- Various ways to execute code (bat, vbs, WMI, PowerShell, compilers, …)

This regardless of the target (workstation, app / Citrix server, ATM, …)

Windows Phone / Mobile exposes only

- Very little information or settings are available
- No interesting default app (you have to download e.g. app «files» separately)
- No possibility to «run» stuff
- No sticky keys
- It’s so impossible to e.g. get the UEFI settings details of the phone…

Application Attack Surface

Microsoft Edge is probably the most interesting app on the phone

«the browser is the new os»

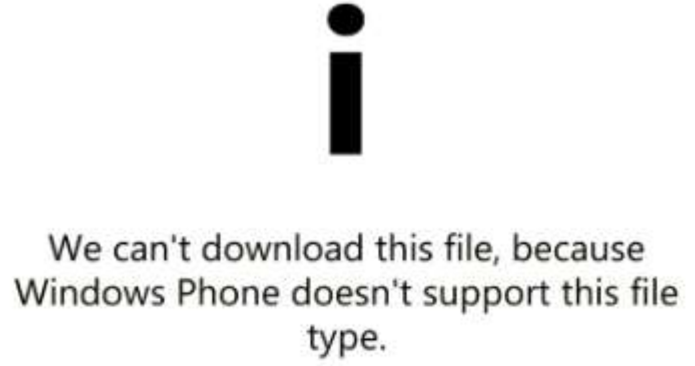
- User influence / interaction
- High privileges on the phone
- Increased attack surface

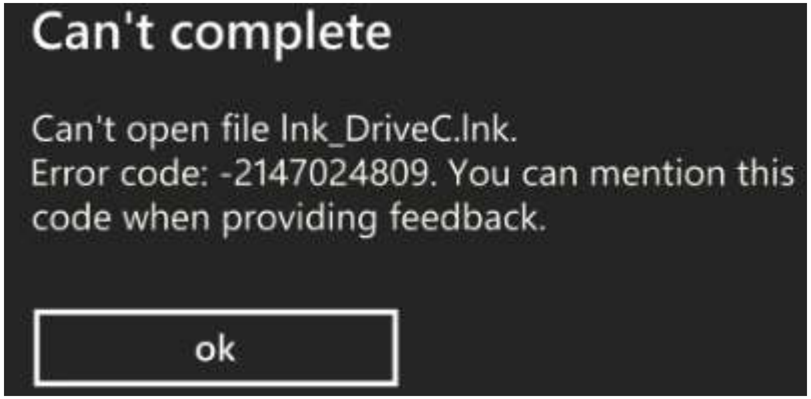
All failed abuse scenarios (so good for the security)

- Run VBScript within the browser
- Browse the local file system using file:///
- SMB connect-back from the phone to the attacker
- No way to download and execute e.g. .bat, .exe, .vbs, … files
- The VB/XSLT <msxsl:script> bypass of Spartan[VB_XSLT] (but crashes when the page is shared)
- Link files (.lnk) are not executed as well…

Application Attack Surface

Microsoft Edge is probably the most interesting app on the phone


We can't download this file, because Windows Phone doesn't support this file type.


Can't complete

Can't open file lnk_DriveC.lnk.
Error code: -2147024809. You can mention this code when providing feedback.

All failed abuse scenarios (so good for the security)

- Run VBScript within the browser
- Browse the local file system using file:///
- SMB connect-back from the phone to the attacker
- No way to download and execute e.g. .bat, .exe, .vbs, … files
- The VB/XSLT <msxsl:script> bypass of Spartan[VB_XSLT] (but crashes when the page is shared)
- Link files (.lnk) are not executed as well…

Development and APIs

If no app provides me the desired feature, let’s code my own!

The C++ and .NET APIs are trimmed down & restricted, preventing breaking out / unwanted actions

All failed abuse scenarios (so good for the security)

- Controlling processes or threats to fork new content within an application
- Running arbitrary commands using Shell.Execute
- Accessing WMI (Windows Management Instrumentation) to gather information and execute arbitrary commands
- Running PowerShell for the same reasons

Open Questions

Of course, not all options have been explored so far, e.g.

- Is arbitrary execution of commands possible, via e.g. Lambda expressions?
- Can the restricted APIs be misused?
  (e.g. attempt to load an assembly not present within the Windows Phone SDK)
- In-depth audit of the Protected Data / Vault feature
- Study of the AppContainer and SIDs separation
- Understand the steps involved in the application signing process
  (and their capabilities restrictions)
- Subversion of accorded capabilities
  (capabilities seem to be labels assigned to a given process).
- Corruption via the video driver e.g. within the browser (WebGL)
- …

Summary

So, what did you actually achieve on this device?

…really not much…

Some information leaks from application crash dumps, e.g.

- User running the app (or let's call it rather the App Container context)
- List of defined drives:
  - C:\
  - D:\ (probably SD card)
  - U:\ (probably a mapping to C:\data\.)
  - PATH variable contains unknown folder C:\WTT\.
  - Data seems shared via C:\Data\Share
  - C:\windows\system32\cmd.exe does not exist

Agenda

Introduction

The Windows View

- Windows Environment
- Attack Surface
- Breaking Out

The Mobile View

- **Sandboxing**
- **Encryption & Enterprise Data Protection**
- **Windows Bridges**

Conclusion

The Mobile View

https://www.microsoft.com/de-ch/mobile/windows10/

Windows Mobile Security Controls

Sandboxing

- Attack Surface Reduction (Least Privilege / Trust Nothing Model)
- User consent and control (Capabilities)
- Isolation (AppContainer, dedicated SIDs, Enterprise Data Protection)

Malware Resistance

- UEFI, Trusted / Secure Boot, Device Guard (System Integrity)
- System and App Integrity (Code Signing)
- Windows Phone Store (Automated Malware Scan)

Exploit Mitigation

- Address Space Layout Randomization (ASLR)
- Data Execution Prevention (DEP)

Encryption

- BitLocker (AES-128 / Customizable, TPM)

Sandboxing

AppContainer

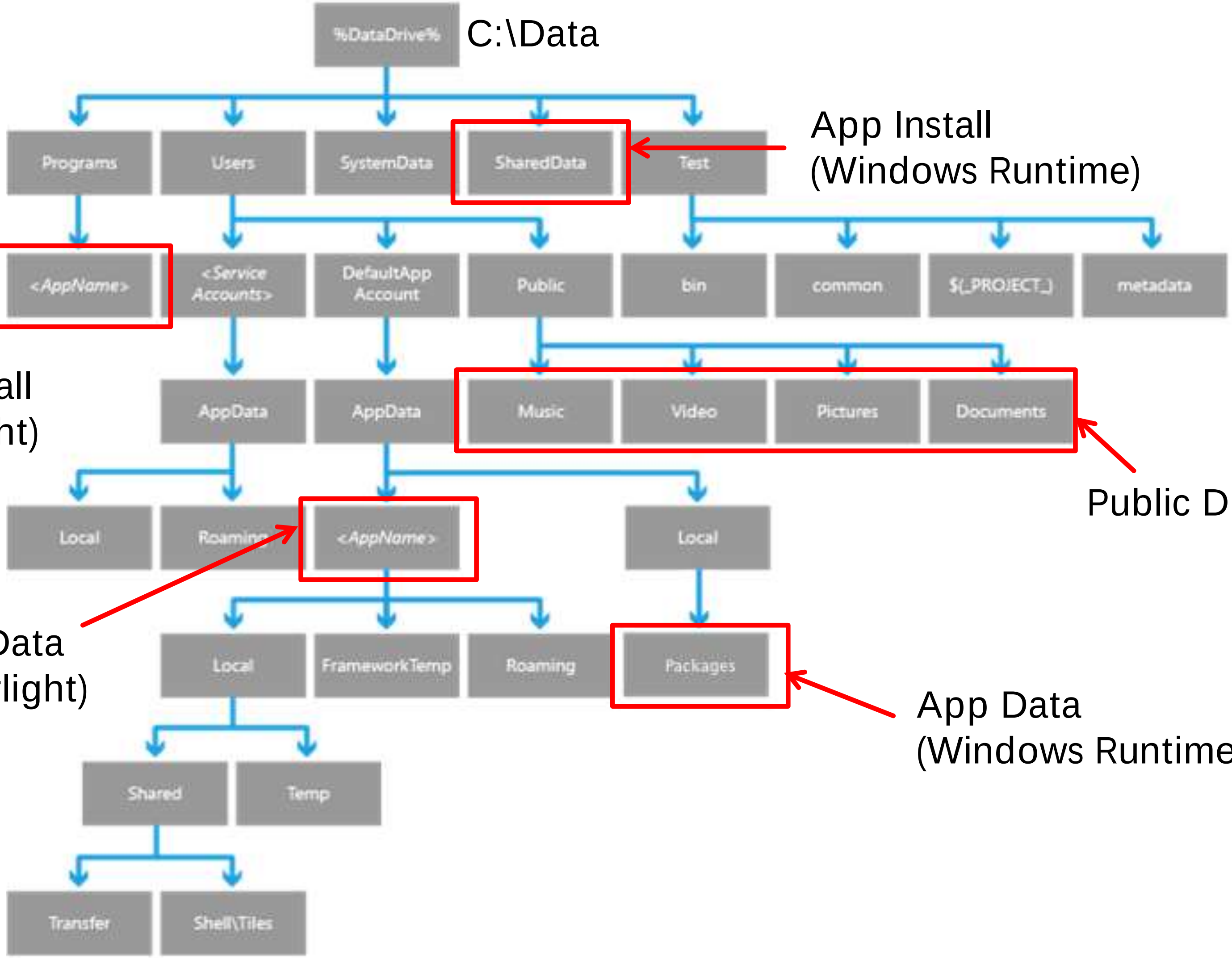
- Isolation
- Data Access
- Credentials
- Roaming
- Sharing Data
- Encrypting data

Capabilities

Restricted APIs

- Isolated Storage / Local Folder

File System Overview



C:\Data

App Install
(Windows Runtime)

App Install
(Silverlight)

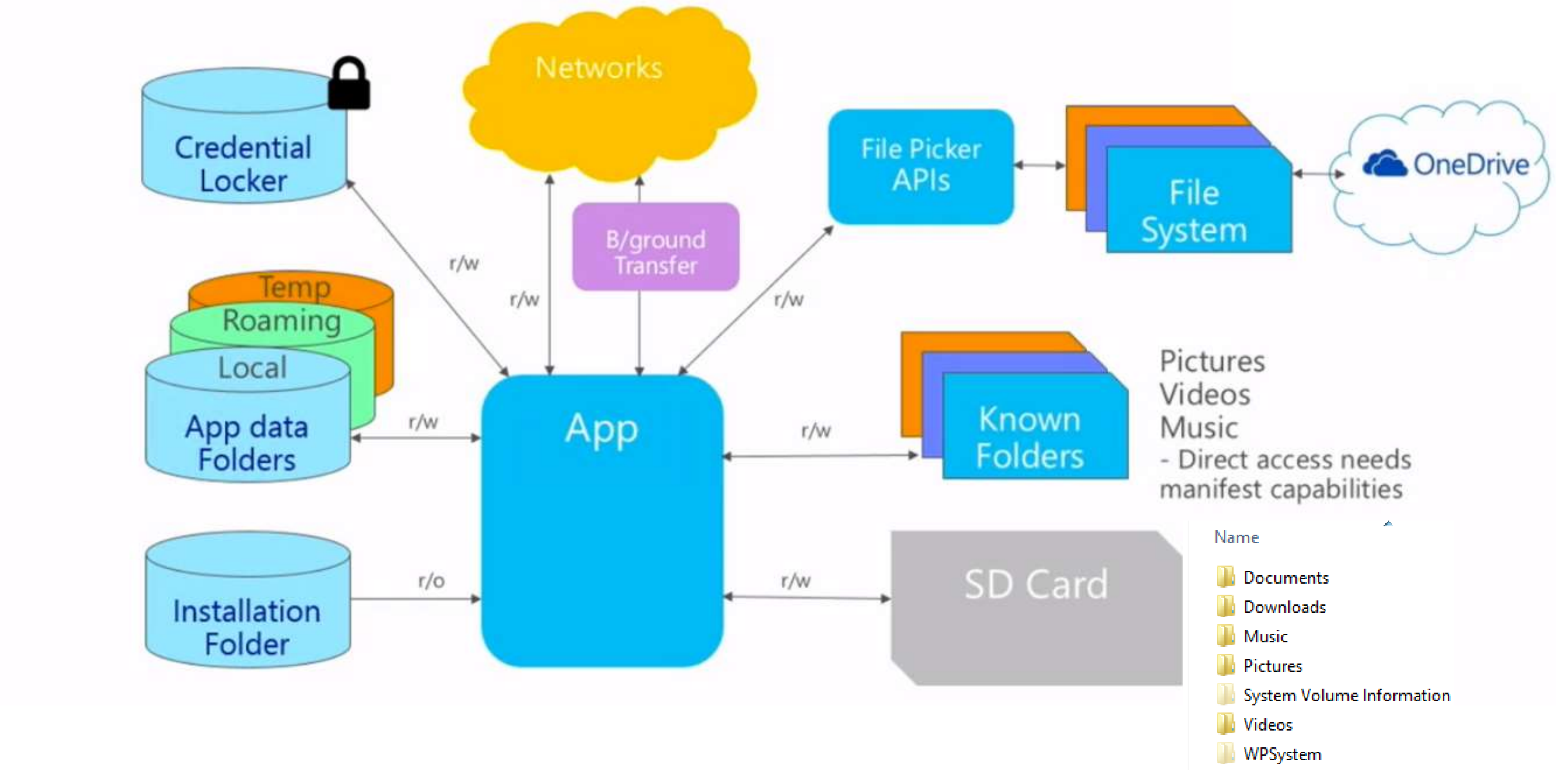
Public Documents

App Data
(Silverlight)

App Data
(Windows Runtime)

Data Access

Locations where apps can access data



Pictures
Videos
Music
- Direct access needs manifest capabilities

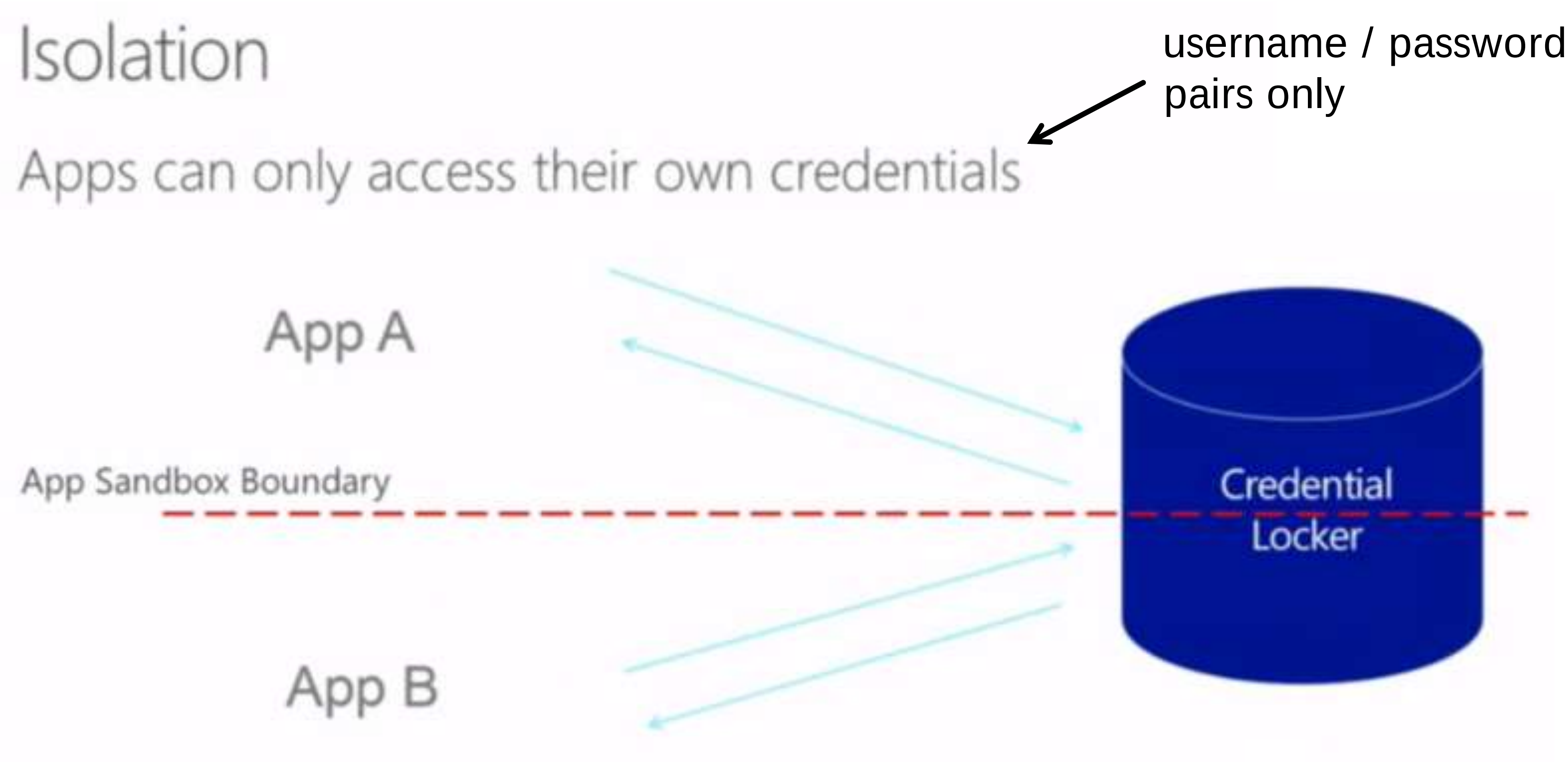
Name

Documents
Downloads
Music
Pictures
System Volume Information
Videos
WPSystem

http://channel9.msdn.com/Series/Building-Apps-for-Windows-Phone-8-1/09

Storing Credentials

Secure Storage & Roaming of Credentials



Isolation

Apps can only access their own credentials

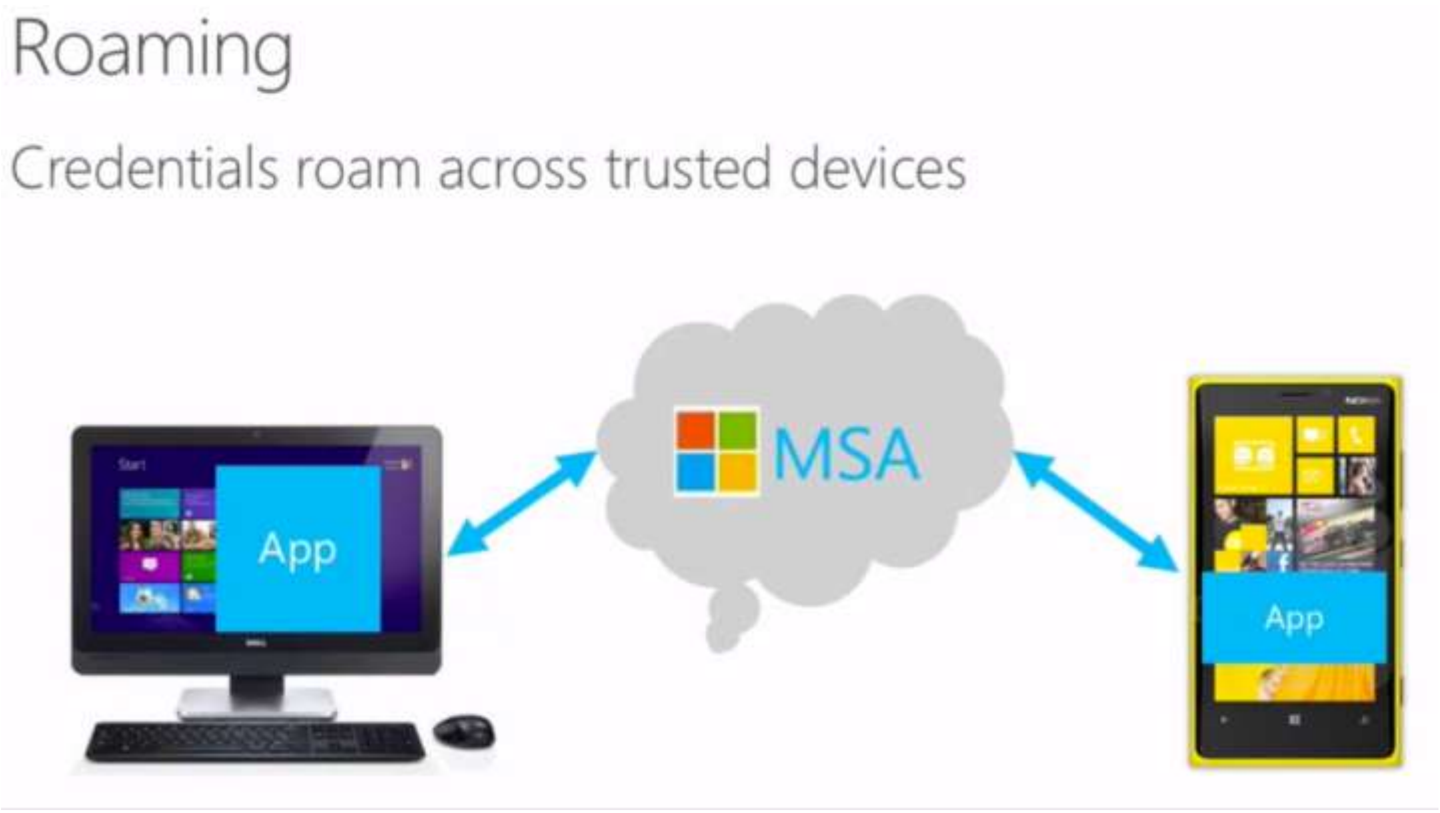
username / password pairs only

Example:

```
var vault = new PasswordVault();
PasswordCredential cred = new PasswordCredential("account", username, password);
vault.Add(cred);
```

Roaming

Sharing data e.g. credentials across devices



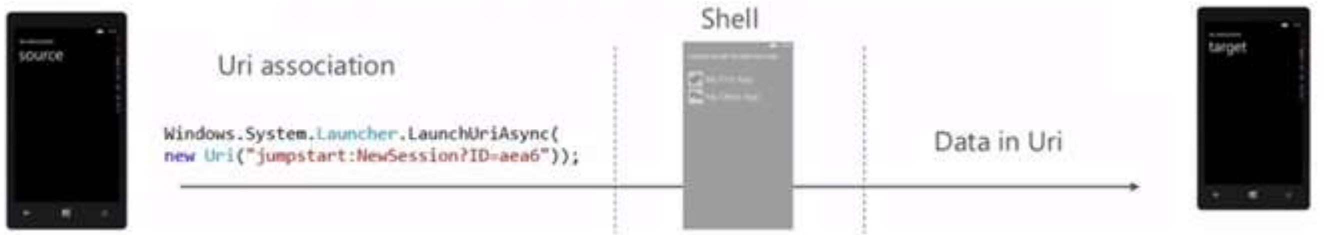
Roaming

Credentials roam across trusted devices

Sharing Data

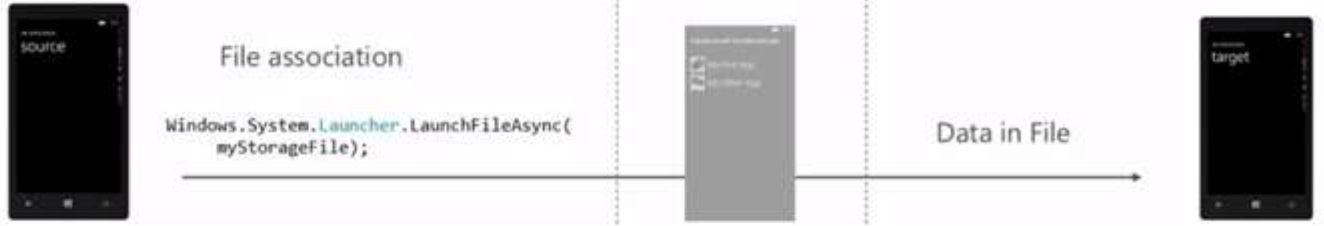
Sharing data between apps works using:

- URI Association, where the registered app obtains the data stored in the URI



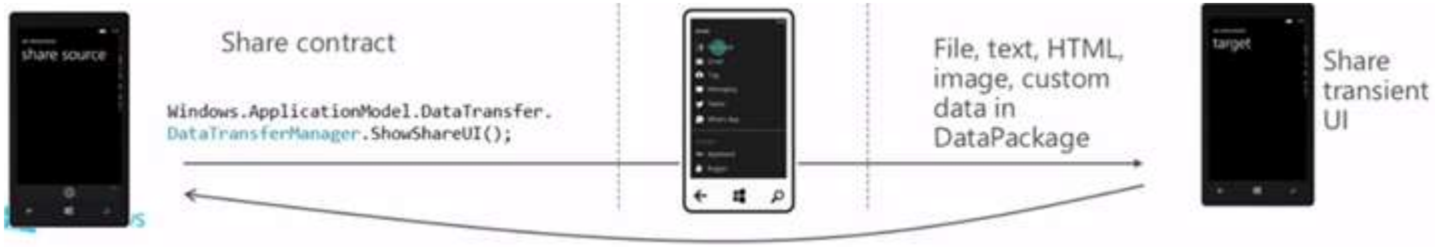
```
Windows.System.Launcher.LaunchUriAsync(
new Uri("jumpstart:NewSession?ID=aea6"));
```

- File Association, where the registered app obtains the file content



```
Windows.System.Launcher.LaunchFileAsync(
    myStorageFile);
```

- Share Contract, allowing custom DataPackages to be shared



```
Windows.ApplicationModel.DataTransfer.
DataTransferManager.ShowShareUI();
```

Encrypting Data

Disk Encryption using BitLocker.

- Since Windows 10 Mobile the end-user can enable encryption.

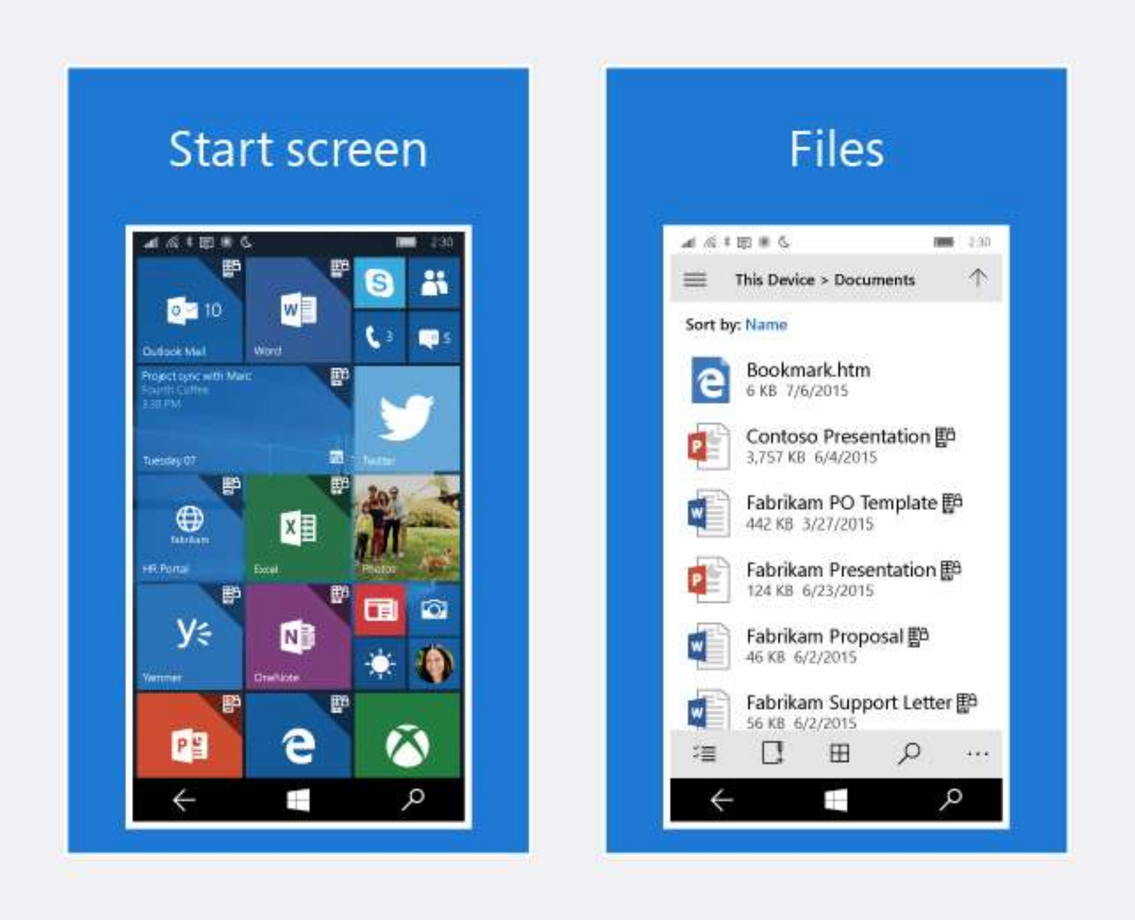
Applications can use DPAPI to protect confidential data.

- DPAPI (Data Protection API) generates and stores a cryptographic key by using the user and device credentials.
- Every app gets it own decryption key, which is created when the app is run for the first time.
- The keys will persist across updates to the app.

Enterprise Data Protection (EDP)

- Automatically tag personal and corporate data
- Protect data while it's at rest
- Control which apps can access corporate data
- Control which apps can access a virtual private network connection
- Prevent users from copying corporate data to public locations

Enterprise Data Protection



Start screen

Files

This Device > Documents

Sort by: Name

Bookmark.htm
6 KB 7/6/2015

Contoso Presentation
3,757 KB 6/4/2015

Fabrikam PO Template
442 KB 3/27/2015

Fabrikam Presentation
124 KB 6/23/2015

Fabrikam Proposal
46 KB 6/2/2015

Fabrikam Support Letter
56 KB 6/2/2015

Capabilities

Software capabilities

- Capability elements are entries in the manifest file that notify the user while installing the app of special software capabilities that your app receives.
- E.g. Provide access to location services

Hardware requirements

- A requirement element is an optional entry in the app manifest file that is used to specify hardware requirements and limit the exposure of an app to users that have a phone with the necessary hardware to run the app.
- E.g. Requiring Near Field Communication (NFC)

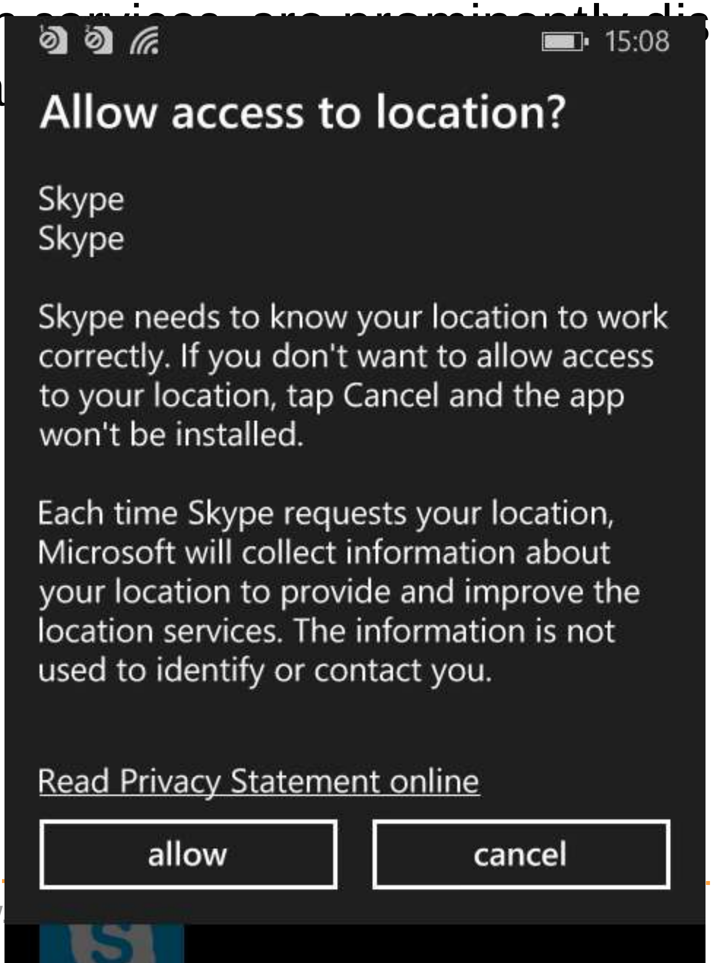
Functional capabilities

- A functional capability is an optional entry in the app manifest file that indicates that your app is requesting a hardware capability of the phone which is present, but not automatically granted.
- E.g. Requesting higher memory limits (in Windows Phone 8.0 only)

Software capabilities

The capabilities listed in the app manifest are disclosed to a user when they view an app for purchase in Windows Phone Store.

Some capabilities, such as location services, are prominently displayed so the user is fully aware that a location information.



Allow access to location?

Skype
Skype

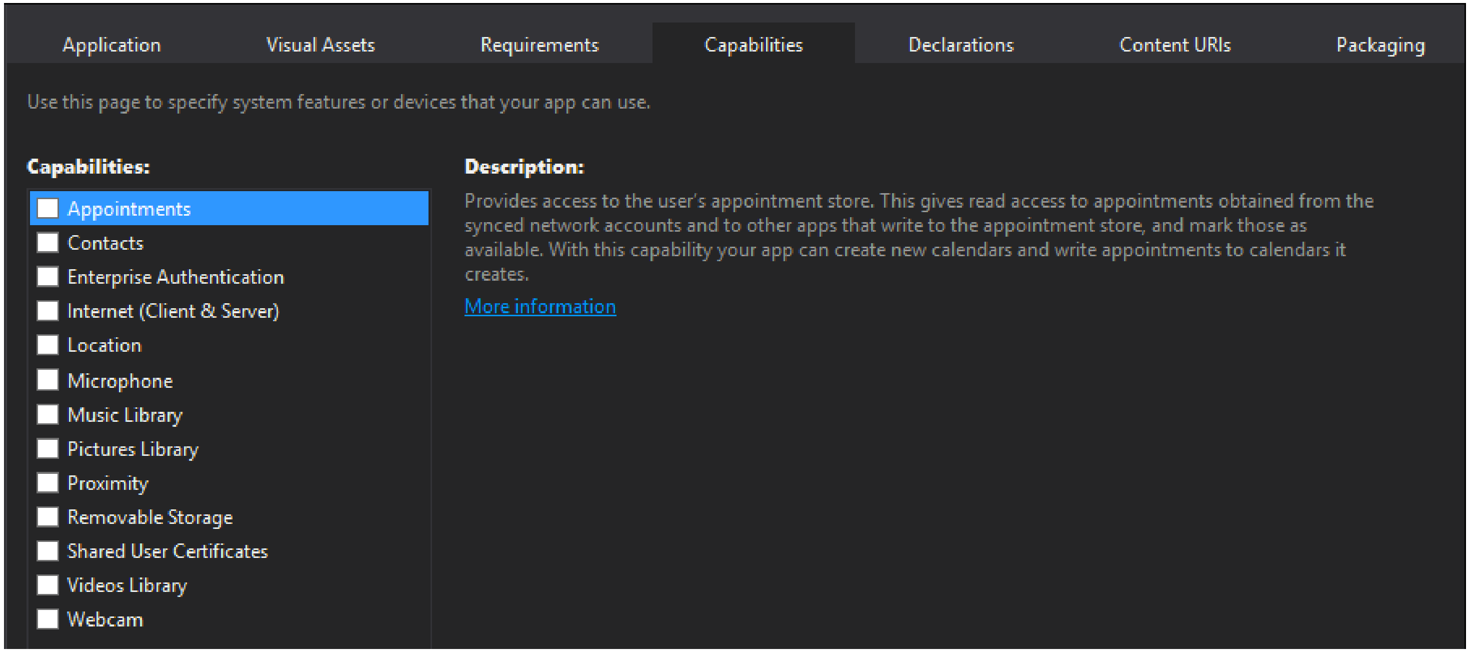
Skype needs to know your location to work correctly. If you don't want to allow access to your location, tap Cancel and the app won't be installed.

Each time Skype requests your location, Microsoft will collect information about your location to provide and improve the location services. The information is not used to identify or contact you.

Read Privacy Statement online

Setting Capabilities

Setting capabilities using Microsoft Visual Studio 2013 Express:



Use this page to specify system features or devices that your app can use.

Capabilities:

- □ Appointments
- □ Contacts
- □ Enterprise Authentication
- □ Internet (Client & Server)
- □ Location
- □ Microphone
- □ Music Library
- □ Pictures Library
- □ Proximity
- □ Removable Storage
- □ Shared User Certificates
- □ Videos Library
- □ Webcam

Description:

Provides access to the user's appointment store. This gives read access to appointments obtained from the synced network accounts and to other apps that write to the appointment store, and mark those as available. With this capability your app can create new calendars and write appointments to calendars it creates.

More information

Note: When testing apps using the Windows Phone emulator the capabilities are granted automatically, even when not included in the app manifest.

Restricted APIs – File Access

Locations all apps can access:

Application install directory

- The folder where your app is installed on the user’s system. (read only)

Application data locations

- The folders where your app can store data. These folders (local, roaming and temporary) are created when your app is installed.

Removable devices (SD Card)

- Access is limited to specific file types

User’s Downloads folder

Restricted APIs – File Access

Locations requiring additional capabilities in the app manifest:

Libraries

- Documents
- Music
- Pictures
- Videos

Removable devices (SD Card)

Homegroup libraries

Media server devices (DLNA)

Universal Naming Convention (UNC) folders

Health Attestation

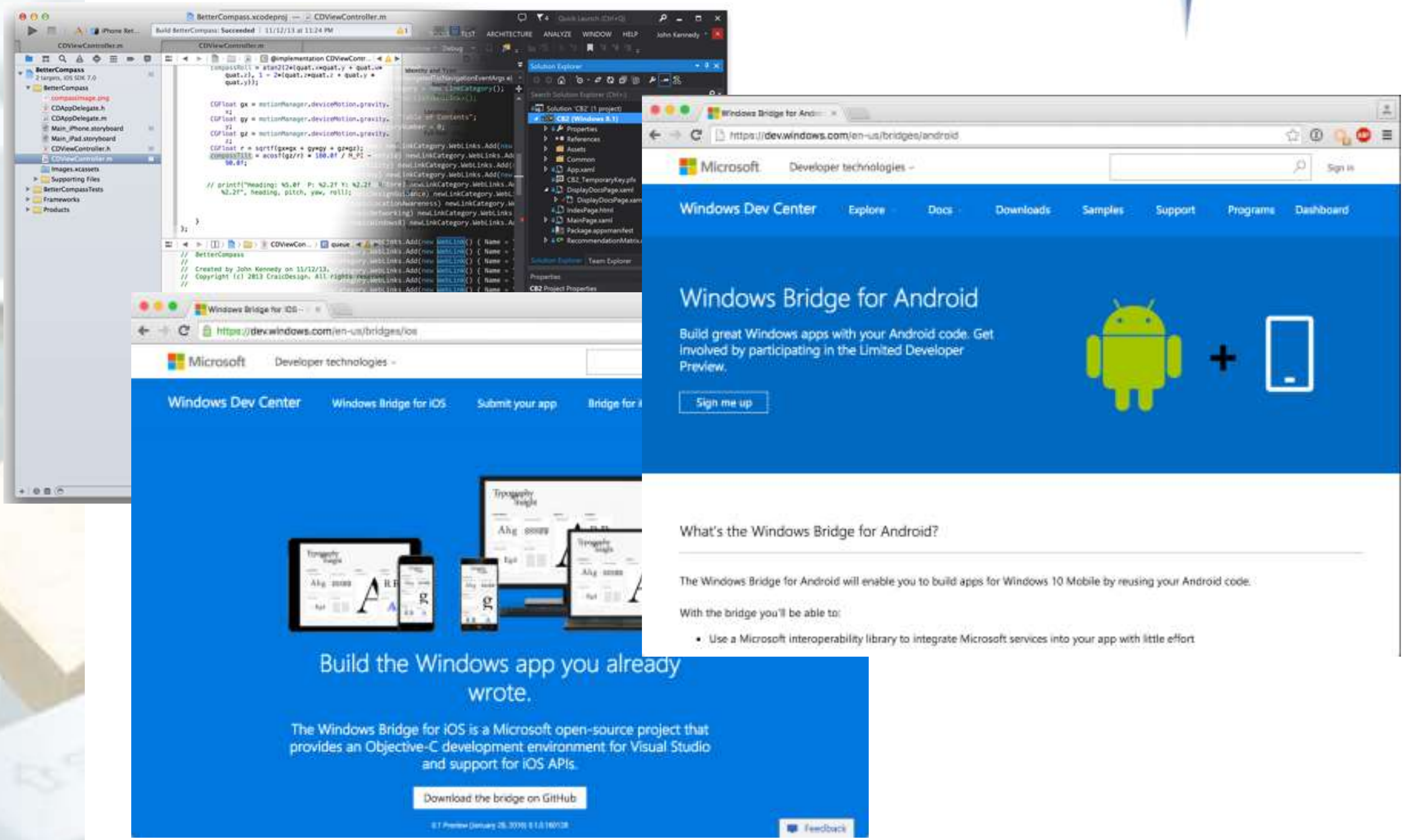
When the user turns on the device a protected and TPM signed audit trail is created.

Devices managed by a health attestation-enabled MDM solution send a copy of the audit trail to the Microsoft Health Attestation Service (HAS).

HAS reviews the audit trail and issues an encrypted and signed report which is then forwarded to the device.

The report can be reviewed using the MDM e.g. to trigger corrective actions.

Windows Bridges



Windows Bridge for Android

Build great Windows apps with your Android code. Get involved by participating in the Limited Developer Preview.

What's the Windows Bridge for Android?

The Windows Bridge for Android will enable you to build apps for Windows 10 Mobile by reusing your Android code.

With the bridge you'll be able to:

- Use a Microsoft interoperability library to integrate Microsoft services into your app with little effort

Build the Windows app you already wrote.

The Windows Bridge for iOS is a Microsoft open-source project that provides an Objective-C development environment for Visual Studio and support for iOS APIs.

Windows Bridges – Current State

Bridge for iOS currently (March 10th) as version 0.1 Preview

- https://dev.windows.com/en-us/bridges/ios
- https://github.com/Microsoft/WinObjC/wiki/Roadmap

Still missing lots of components

The Windows Bridges for Android project has been cancelled recently

Xamarin (Cross-Platform Development) bought in February 2016

Wi-Fi Sense

Automatically connects you to Wi-Fi networks around you.

- Open Wi-Fi networks known by crowdsourcing e.g. other Windows Phone users have connected to.
- Accept the Terms of Use on your behalf.
- Provide additional information such as e-mail address or phone number on your behalf. (In some countries generic info will be used by default)
- Shares your Wi-Fi credentials with your Facebook friends, Outlook.com or Skype contacts.

Can I prevent my users from sharing their credentials?

- Yes, if you don't mind adding "_optout" to your Wi-Fi SSID
- What about Google's "_nomap" suffix then?

https://www.windowsphone.com/en-us/how-to/wp8/connectivity/wi-fi-sense-faq

Conclusion

Windows Phone / Mobile

- Windows 10 Mobile has just been released on March 17th 2016
- Some devices using Windows Phone 8.1 can be updated
- Automatic Updates for Windows 10 Mobile
- Windows 10 Mobile Enterprise users can postpone updates
- Biometrics (Windows Hello) limited to select premium models

- Is more a phone similar to iOS and Android than a Windows desktop
- Is based on secure and proven good security technologies

Questions?