Windows RunTime

Hack In The Box 2012

Sébastien RENAUD srenaud@quarkslab.com
Kévin SZKUDLAPSKI kszkudlapski@quarkslab.com

Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

How it's started

- Searching for something new in Windows 8
- Let's see what's new in the Kernel!
- Diffing Windows 7 RTM Kernel vs. Windows 8 DP Kernel
- Stumbled across `NtCreateLowBoxToken()`
- Unwinding the thread: Windows Runtime (WinRT)!

Metro & WinRT

- Windows 8 new interface: Metro
- Metro style apps (aka immersive apps)
- WinRT: Backbone of Metro apps / new programming model

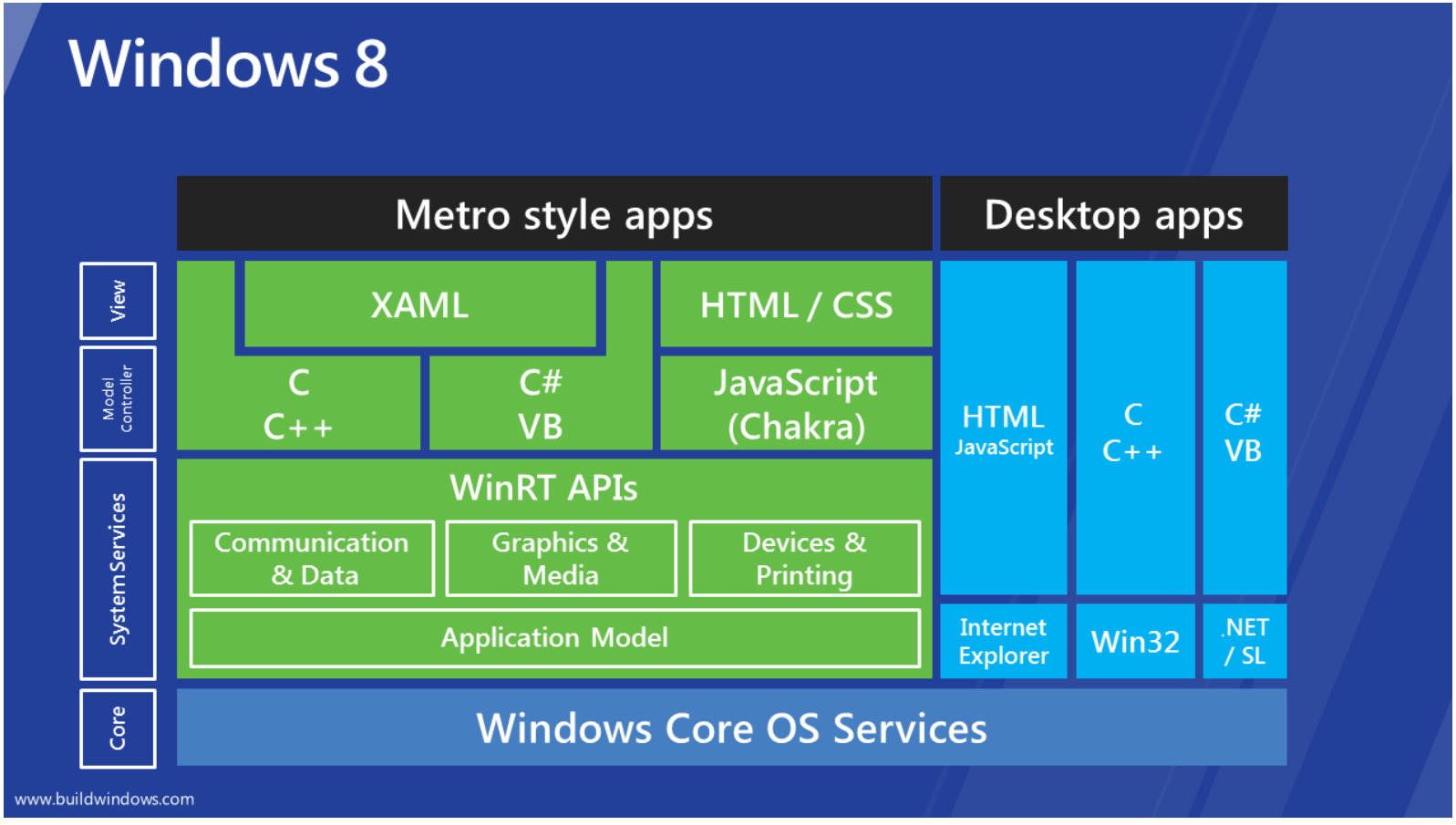
Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

Metro Apps: Keys points

- Distributed only through the Windows Store
- Executed in an "App Container"
  - Secured through a sandbox
  - Severly limited resources access
  - Limited resource access: need explicit permissions
  - Use a restricted subset of .NET and Win32 APIs

WinRT: Big picture



Application Package

- Applications are installed per user
- Application are packaged (*.appx) for deployment
  - Package is signed
  - Package is compressed
  - Contains all needed files
  - Can target multiple plateforms (x86; x64; ARM)

Application Installation

- Only through the Windows Store
- AppxManifest.xml describes application registration

Registration

- <Application>...</Application>: core of the registration
- <Capabilities>...</Capabilities>: What am I allowed to do
- <Extensions>...</Extensions>: What can I use

Everything is mapped onto the registry (HKCU).

Capabilities

Capabilities

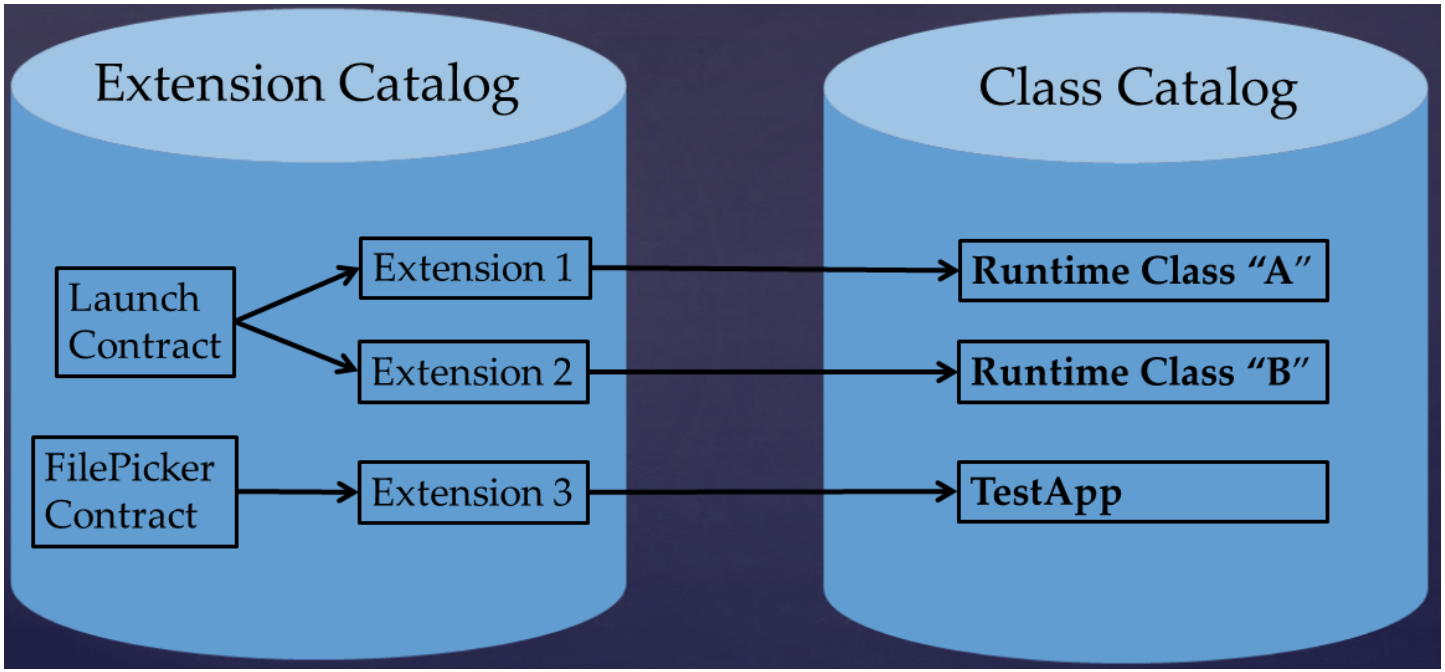
- Network: Enterprise auth., client, server & client, Intranet, Text Messaging, etc.
- File System: Documents, Pictures, Music, Video, etc.
- Devices: Location (e.g. GPS), Microphone, Proximity (e.g. NFC), Removable storage, etc.

Things that are specific to an application (local storage, settings, etc.) do not require capabilities.

Class and Extension

Catalogs

- Extension: "I implement this contract" (e.g. Launch).
- Class: describes the WinRT classes (implementation).



Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

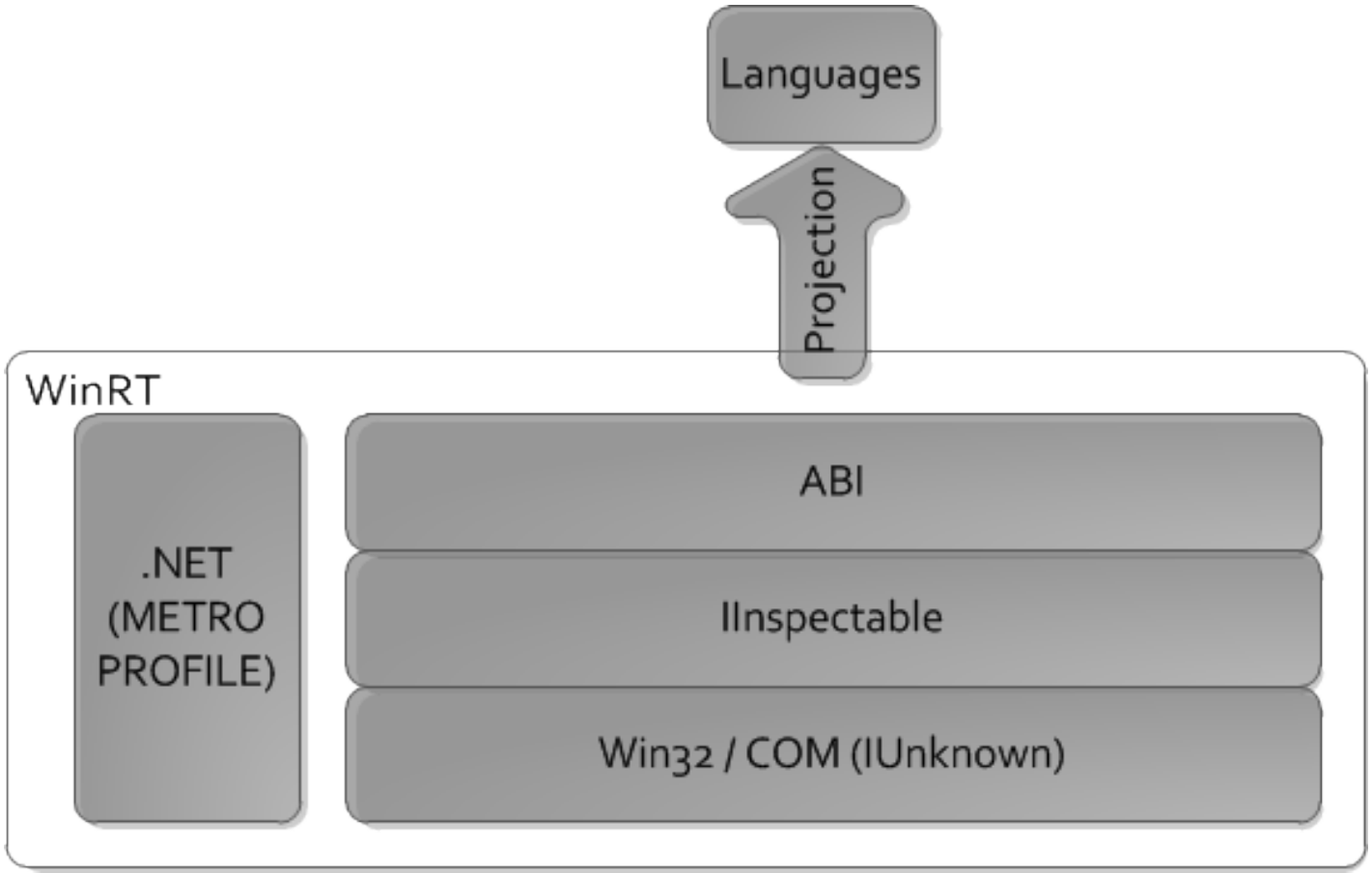
Application startup

Application automatically implements the "Launch contract".

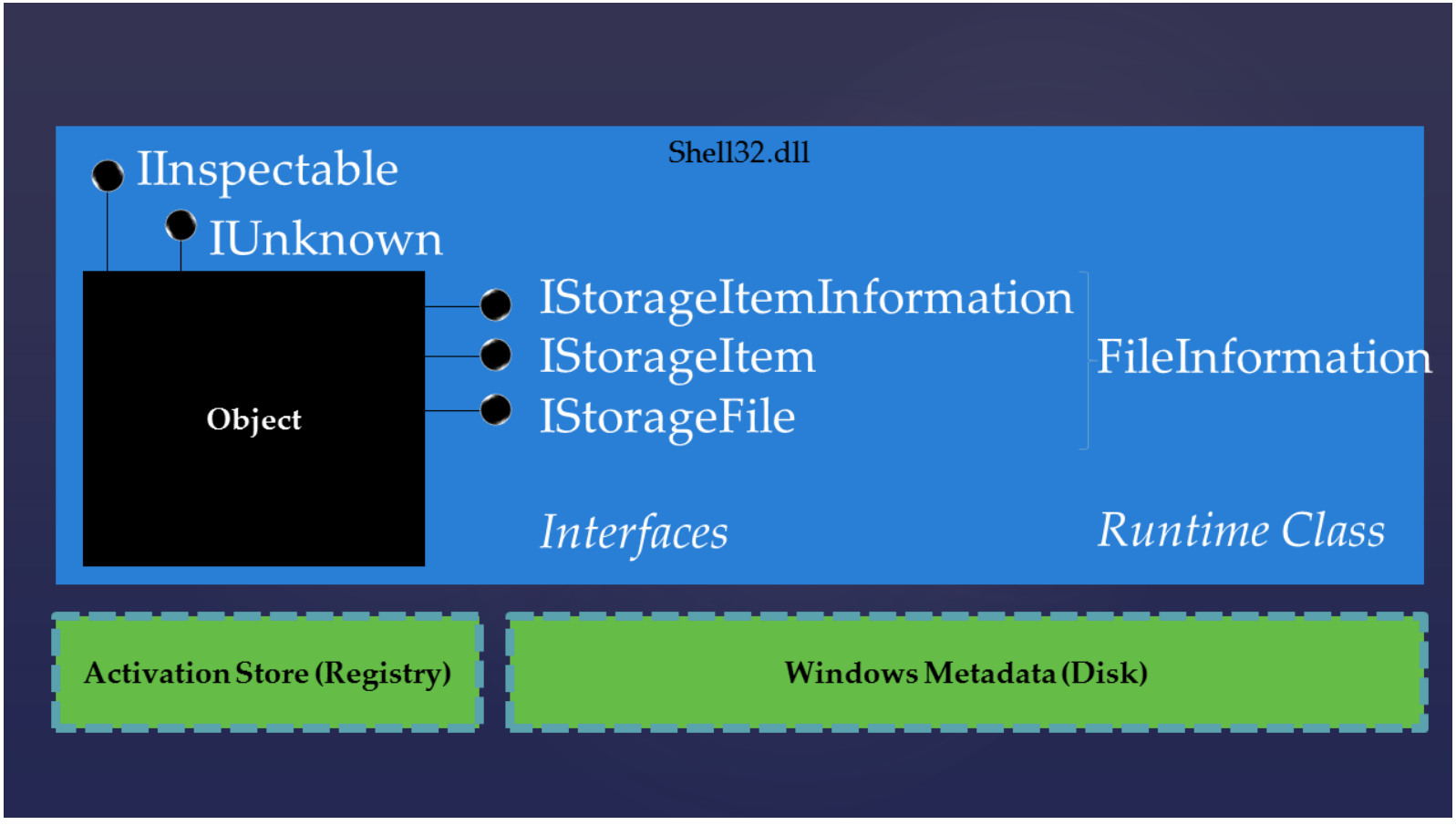
App startup: key points

- System queries the extension catalog to find the right extension
  - Explorer.exe queries the extension catalog
  - Check if it's the right object to activate
  - Activate the object
- Activation
  - Send request to RPCSS
  - Is the process already running?
  - If not already running, send request to DCOM Launch service
  - Start the application

WinRT: base


Languages
Projection
WinRT
.NET (METRO PROFILE)
ABI
IInspectable
Win32 / COM (IUnknown)

WinRT: Object example



Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

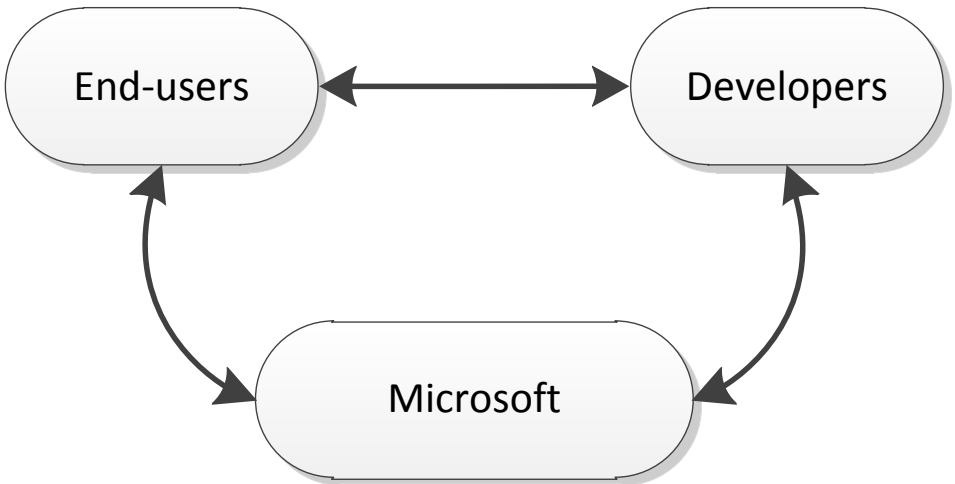
Purpose

- Unique way to download winrt applications
- Microsoft controls all applications (signature is mandatory)
- Applications checking:
  - Must be linked with SAFESEH, DYNAMICBASE and NXCOMPAT
  - Must not hang or crash
  - List of forbidden API

API list checking by "Windows App Certification Kit"

- Checking is done statically
- Can be bypassed by retrieving API address dynamically (shellcode technique)

Windows 8 Ecosystem



AppContainer

- AppContainer, new sandbox concept
- Defined a list of capabilities per application
- New flag in PE header

```
1  // _IMAGE_OPTIONAL_HEADER::DllCharacteristics
2  #define IMAGE_DLLCHARACTERISTICS_APPCONTAINER 0x1000
```

Capabilities

SID	Name
S-1-15-3-1	Your Internet connection
S-1-15-3-2	Your Internet connection, including incoming connections
S-1-15-3-3	A home or work network
S-1-15-3-4	Your pictures library
S-1-15-3-5	Your videos library
S-1-15-3-6	Your music library
S-1-15-3-7	Your documents library
S-1-15-3-8	Your Windows credentials
S-1-15-3-9	Software and hardware certificates or a smart card
S-1-15-3-10	Removable storage

Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

Sandbox

What is a sandbox ?

A sandbox is a mecanism to isolate untrusted processes.

What does a sandbox contain ?

- Isolated process which runs with very limited rights
- Broker, a process which could execute specific actions for a isolated process
- An IPC mecanism to allow isolated processes to communicated with broker

Sandbox on Windows

- Restricted token
- Job
- Desktop / WinStation
- Low integrity level (since windows vista)

Sandbox on Windows

- Restricted token
  - `CreateRestrictedToken` or `NtFilterToken`
  - Disable or restrict SID
  - Delete privileges
- Job
- Desktop / WinStation
- Low integrity level (since windows vista)

Sandbox on Windows

- Restricted token
- Job
  - `CreateJobObject` / `AssignProcessToJobObject`
  - Limit acccess to desktop, clipboard, global hook, atom table, ...
  - Forbid the creation of a sub process
  - Restrict the use of CPU, memory and IO
- Desktop / WinStation
- Low integrity level (since windows vista)

Sandbox on Windows

- Restricted token
- Job
- Desktop / WinStation
  - `CreateDesktop(Ex)`
  - Windows message isolation
  - Clipboard, Atom, ... can be isolated too
- Low integrity level (since windows vista)

Sandbox on Windows

- Restricted token
- Job
- Desktop / WinStation
- Low integrity level (since windows vista)
  - `SetTokenInformation`
  - Read access in filesystem or registry unchanged
  - Only write access to folder *"%UserProfile%\AppData\LocalLow"* and registry *"HKEY_CURRENT_USER\Software\AppDataLow"*
  - User Interface Privilege Isolation forbids to send "write"-type message to higher level integrity process
  - Can't change privileges
  - ...

Sandbox on Windows

- Restricted token
- Job
- Desktop / WinStation
- Low integrity level (since windows vista)

Limitation

- No way to forbid a process to call syscall (like seccomp)
- Some object can't be secured (fat fs)

Chrome vs. WinRT

Why Chrome ?

- Windows sandbox implementation
- Open source and well documented
- Designed for security only (contrary to AppContainer)

Comparaison points

- Process isolation
- Broker process
- Sandbox communication

Process isolation

Chrome

- RESTRICTED SID (S-1-15-2) is set to restricted
- Most of SID group are disabled
- Isolation relies on job and
  - (on Windows XP) desktop
  - (on Windows Vista and superior) integrity level
- Has to call `TargetServices::LowerToken` to be isolated

LowBox

- Microsoft modified `_TOKEN` structure
- A new syscall `NtCreateLowBoxToken` to make a very limited token
- `SepAccessCheck` was slightly modified

Process isolation

Chrome

...

LowBox

- Microsoft modified `_TOKEN` structure
  - PackageSid (unique per application)
  - CapabilitiesSid
  - Lowbox number entry
  - Handle (?)
  - New _TOKEN::Flags TOKEN_IS_IN_APP_CONTAINER (0x4000)
- A new syscall `NtCreateLowBoxToken` to make a very limited token
- `SepAccessCheck` was slightly modified

Process isolation

Chrome

...

LowBox

- Microsoft modified `_TOKEN` structure
- A new syscall `NtCreateLowBoxToken` to make a very limited token
  - Fills new fields
  - Sets integrity level to low
  - Changes access rights to the token to TOKEN_ALL_ACCESS for itself and TOKEN_QUERY for administrators
- `SepAccessCheck` was slightly modified

Process isolation

Chrome

...

LowBox

- Microsoft modified `_TOKEN` structure
- A new syscall `NtCreateLowBoxToken` to make a very limited token
- `SepAccessCheck` was slightly modified
  - Checks if _TOKEN::Flags & TOKEN_IS_IN_APP_CONTAINER (0x4000)
  - (Current theory) add a new test: accessed object must contain either the current *PackageSid* or the well-known SID *"ALL APPLICATION PACKAGES"*

Broker

Chrome

- Broker process and sandboxed processes are the same executable on disk (chrome.exe)
- `sandbox::SandboxFactory::GetBrokerService` is used to differentiate (fork() style)
- Implements its own access policies system

LowBox

- COM interface (RuntimeBroker.exe)
- Automatically run by `svchost.exe`
- `CoImpersonateClient` is used to retrieve sandboxed process token
- `RtlCheckTokenCapability` is called to test sandboxed process access

Inter-process communication

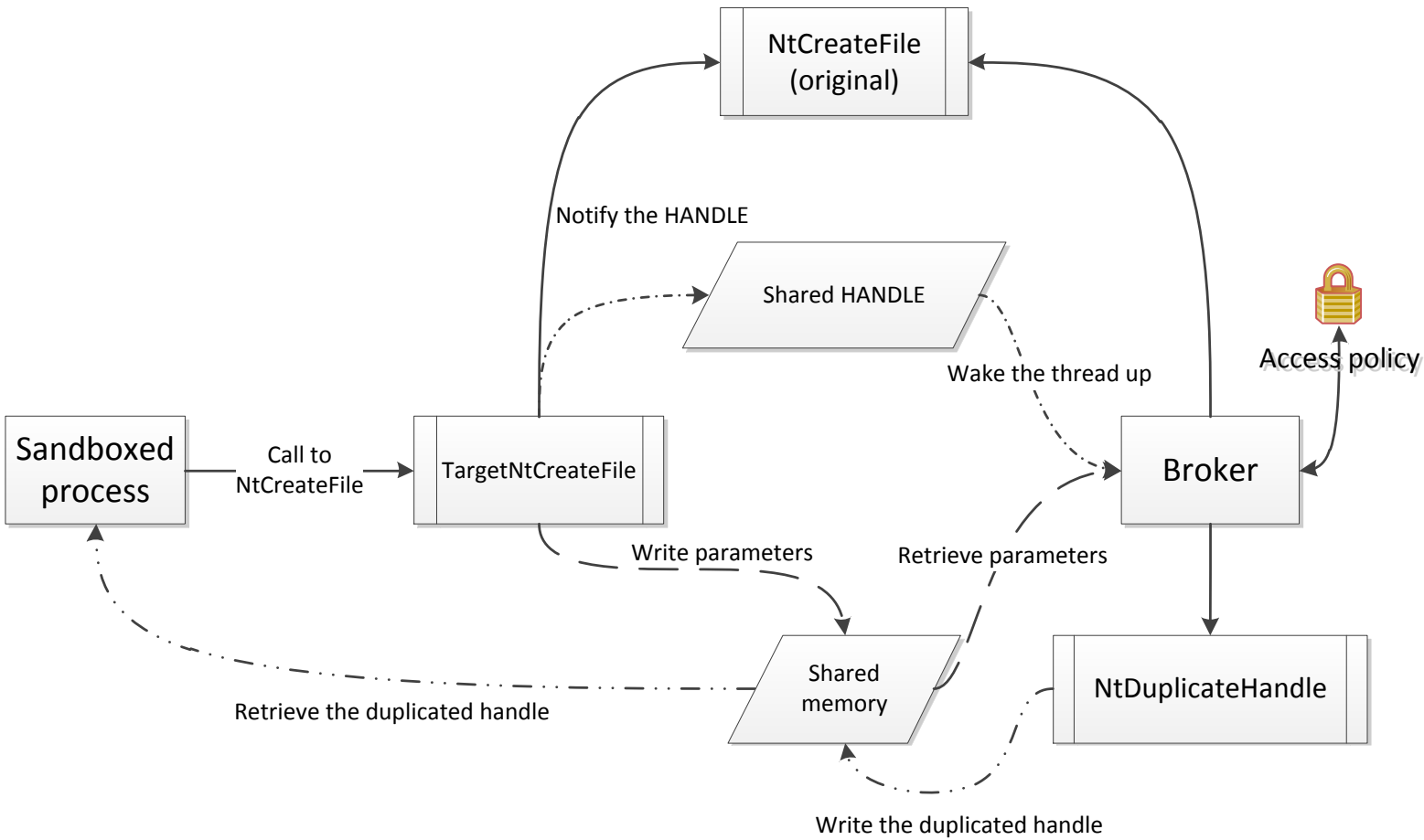
Chrome

- API hooking used to easily sandbox process (closed source plugin)
- Shared memory is used to transport parameters / result
- Duplicated handle is used by the sandbox to wake the broker up

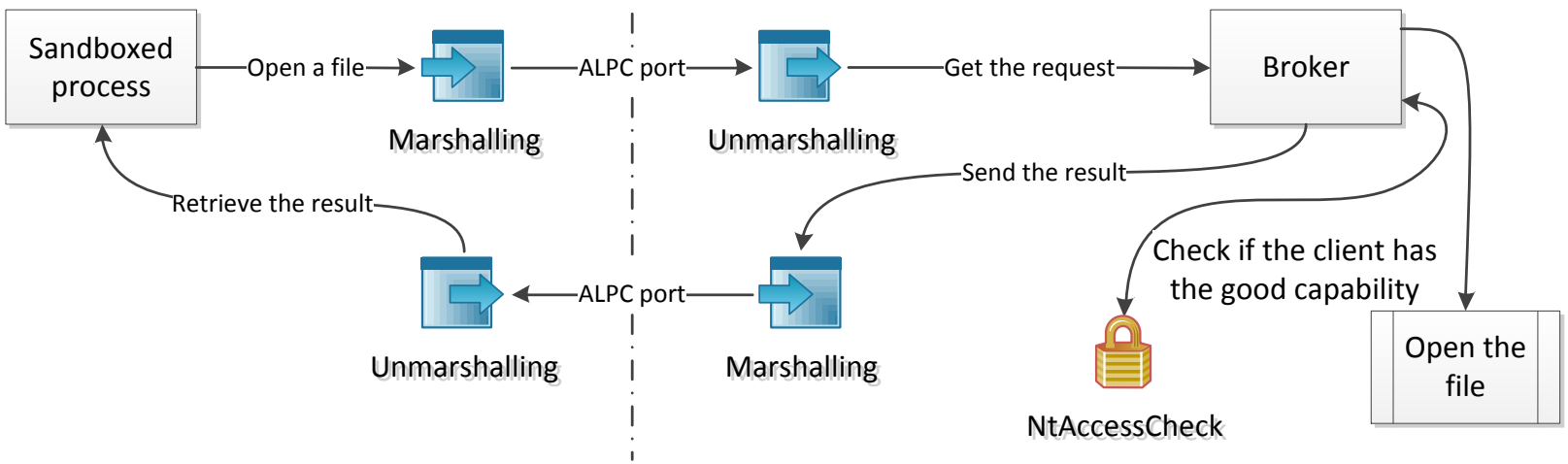
LowBox

- Relies on COM
- Each request is a COM object
- Uses an ALPC port to transport marshalled COM object (`NtAlpcSendWaitReceive`)

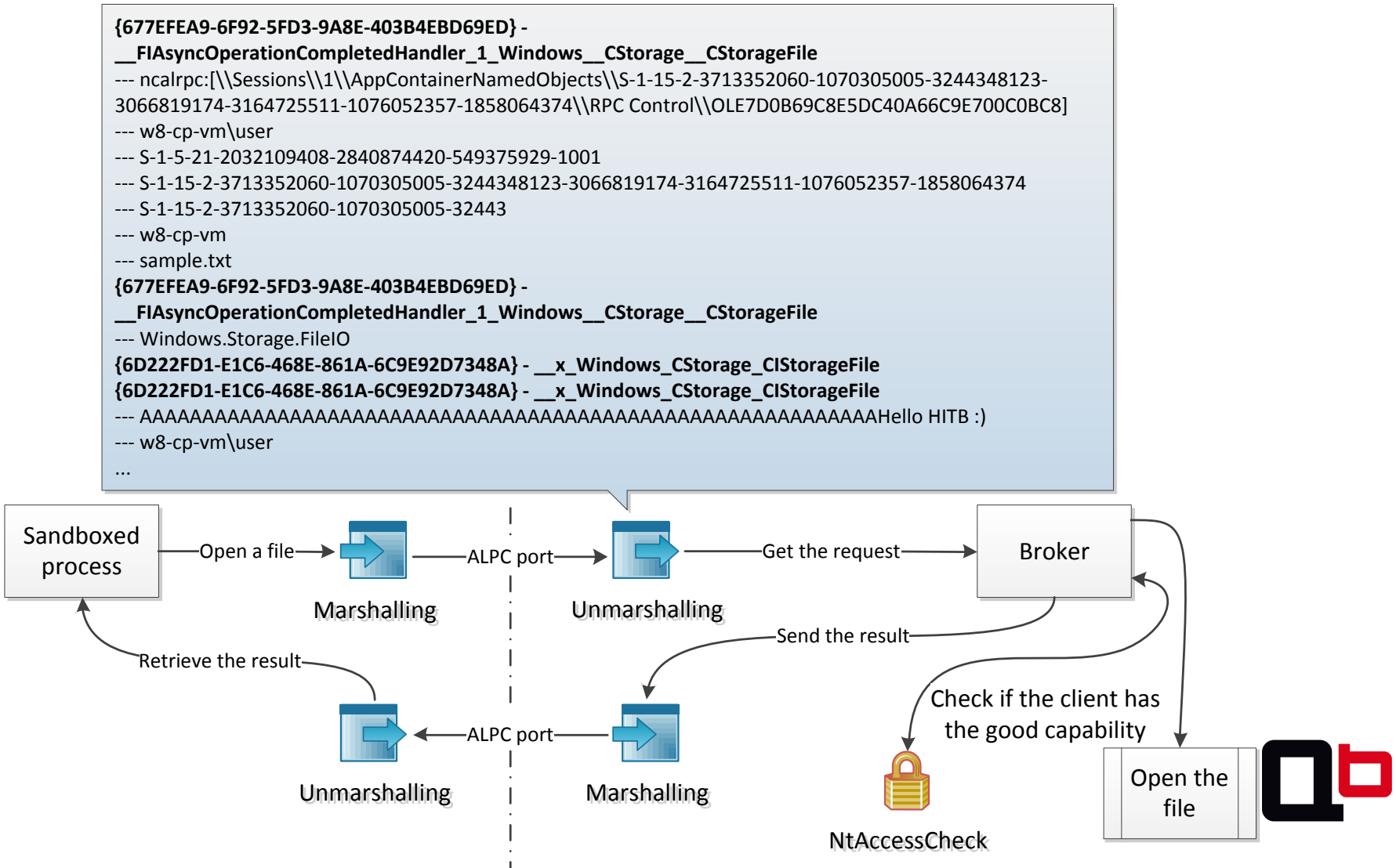
Chrome sandbox - Layout



WinRT sandbox - Layout



WinRT sandbox - Layout + hook



Plan

1. Windows 8
2. WinRT - Applications & Components
3. WinRT - Internals
4. Windows Store
5. Sandbox
6. Conclusion

Conclusion

WinRT

- New design
- New API
- Mainly based on COM

AppContainer

- Provide some level of isolation
- Transparent to users / developers
- Isolation implemented in kernel

Thanks

- The QB team
- Microsoft
- The HITB team

Questions?

www.quarkslab.com

contact@quarkslab.com | @quarkslab.com