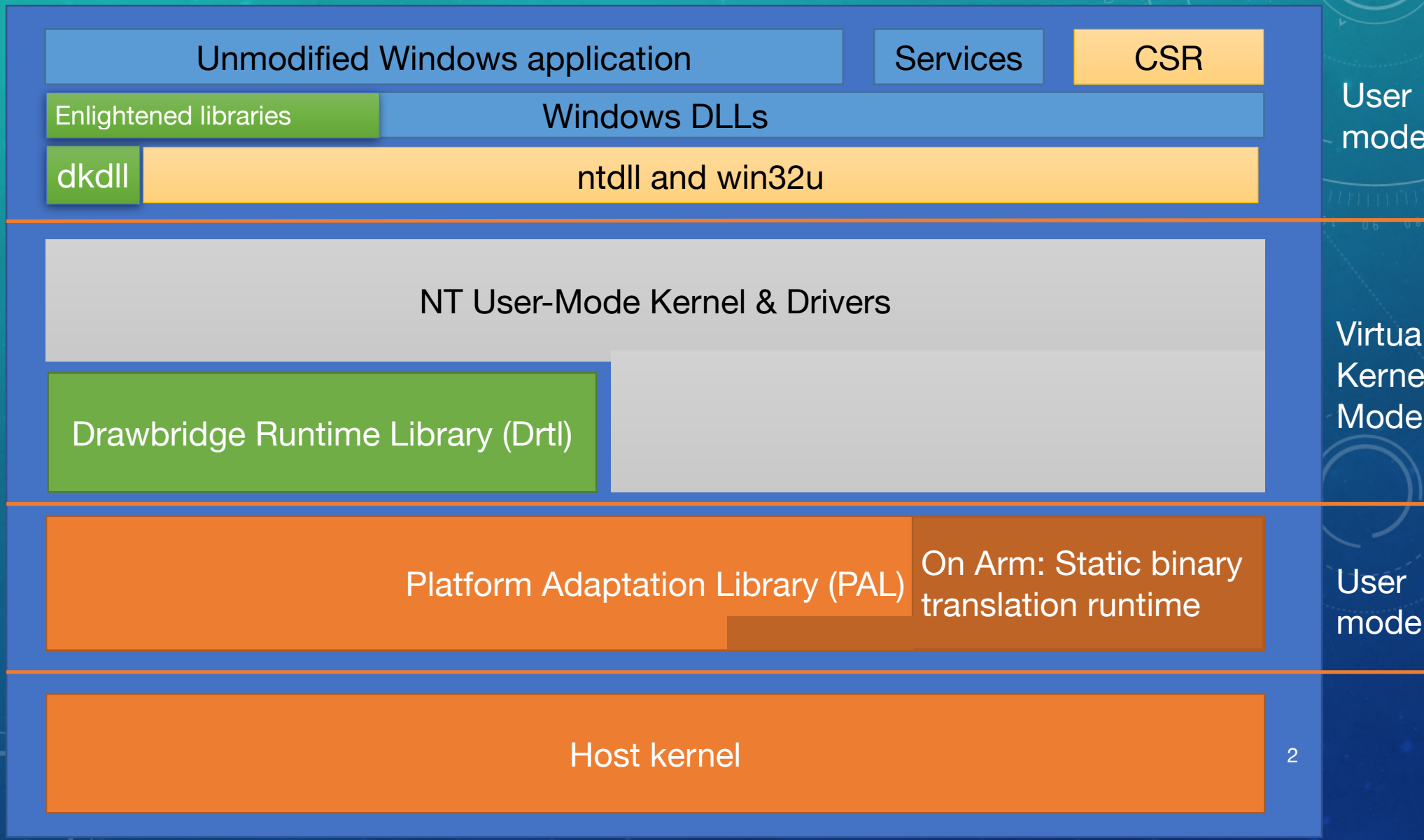
DRAWBRIDGE

Documenting a bit something that is totally undocumented…

A HIGH LEVEL OVERVIEW



INTRODUCTION

- Drawbridge is at the heart of the SQL Server on Linux project, allowing very significant code sharing advantages. *(of course not everything, this is a database engine and AIO semantics aren't that compatible)*
- Microsoft is improving the Drawbridge libOS significantly as a part of the SQL Server engineering effort, and upgraded it from Windows 8.0 to Windows 10 rs4.
- The PAL and the LibOS can be updated independently of the application.

DRAWBRIDGE: THE API

DRAWBRIDGE API

Memory management:

VirtualMemoryAllocate
VirtualMemoryFree
VirtualMemoryProtect

Events:

EventClear
EventPeek
EventCreateNotificationEvent
EventSet
NotificationEventCreate

Semaphores:

SemaphoreCreate
SemaphorePeek
SemaphoreRelease
SemaphoreReleaseEx
SynchronizationEventCreate

Threads:

ThreadCreate
ThreadInterrupt
ThreadExit
ThreadSetRegisters
ThreadSetAffinity
ThreadYieldExecution
ThreadAssertAffinity

Streams:

AsyncCancel
AsyncPoll
AsyncCancelPumpIoRequest
StreamAttributesQuery
StreamAttributesQueryByHandle
StreamChangesPoll
StreamChangesRegister
StreamDelete
StreamEnableSparse
StreamEnumerateChildren
StreamFlush
StreamMap
StreamMapPeBinary
StreamOpen
StreamQueryAllocatedRanges
StreamRangeLock
StreamRangeUnlock
StreamRead
StreamReadScatter
StreamRename
StreamSetLength
StreamSetZeroData
StreamUnmap
StreamGetEvent
StreamWrite
StreamWriteGather
StreamControl
StreamEventSelect
StreamEventEnum

GUI:

ConsoleCreate
ConsoleEventPoll
ConsoleNotifyUpdate

DRAWBRIDGE API (CONTINUED)

Process:

ProcessCreate
ProcessExit
ProcessGetExitCode
ProcessTerminate

Objects:

ObjectClose
ObjectReference
ObjectWaitAny

Other:

ExceptionRecordFree
RandomBitsRead
SystemTimeQuery
Syscall
SystemCpuUtilizationQuery

Packets:

PacketCreate
PacketAssociate

Debug:
DebugStringPrint
Debug_NotifyLibOSBooted

Enclaves:

Enclave_CreateSgx
Enclave_InitializeSgx
Enclave_CreateVbs
Enclave_InitializeVbs
Enclave_LoadData
Enclave_LoadPeBinary
Enclave_PagesCommit
Enclave_PagesFree
Enclave_PagesProtect
Enclave_CallVbs
Enclave_Delete
Enclave_PagesRemove
Enclave_NotifyThreadCreate

Network:

NetworkIoPort_Open
NetworkIoPort_Pump
NetworkIoPort_Wakeup

Upcalls:

LibOsInitialize
LibOsExceptionDispatch
LibOsThreadStart

I/O: SUPPORTED PROTOCOLS

I/O:

stdin:
stdout:
stderr:
null:
clock:
signal:
protectdata:
asyncio:
networkio:
affinity:
kerberos:
symmetrickey:
ssl:
pam:

Named pipes:

pipe.server://
pipe.client://

Shared memory:

shm:
streamsemaphore:

DRAWBRIDGE API

- The API has evolved significantly since the one shown in the research papers, and continued to evolve between SQL Server 2017 and 2019.
- Stream API is affected the most, new specific network APIs instead of only streams too.

DRAWBRIDGE: THE ABI

NTABI_THUNK & MORE

ENTRY POINT

- LibOSInitializeOnStack(DK_THREAD_START_PARAMETERS* StartParameters,DK_LIBOS_INIT_DATA* InitData)
- StartParameters points to Windows-specific LibOS settings. It's the loader block.
- InitData contains the settings that are independent from the target LibOS. It also contains the pointer to the upcall & call tables.
- Small stack that the LibOS switches from as soon as possible.

DRAWBRIDGE ABI: GUEST TO PAL

- Before: DK_CALL_TABLE, now returns “Unexpected call to legacy ABI.” when used (except for Syscall).

  SQL Server 2017’s PAL uses the legacy ABI while SQL Server 2019 has the new one.

- Now: AbiTableGetFunction, also named as DKAbiGetFunction in NTUM. The Syscall function is used for calling it (offset 0x1D0 to DK_CALL_TABLE).

SYSCALL MARSHALLING

- The infrastructure is currently mostly unused, with the unmarshaling table only covering two functions, Abi_GetFunction_v2 and Abi_GetVersion_v2.
- Otherwise, the syscall marshalling path isn't currently used at all.
- Syscall marshalling is new to SQL Server 2019's Drawbridge implementation, before that point, it was just a stub.

GETTING A FUNCTION FROM THE DIRECT ABI CALL TABLE

- Input size for Abi_GetFunction is 4 bytes to identify the ABI function & the function ID. Output size is 8 bytes, the size of the thunking pointer to be returned.
- This is clearly a work in progress, perhaps to be changed in depth later on.

THE PAL

WHAT ARE THE DIFFERENCES AND CONSTRAINTS SPECIFIC TO THE DRAWBRIDGE LINUX PAL?

THUNKING

- Thunking is present because of incompatible ABIs between Windows & Linux.
- Calls from NTUM to the PAL go through ntabi_thunk for adaptation purposes. This isn’t the only possible design, but allows to decrease complexity.

MEMORY PROTECTION KEYS

- Why? Unlike user-mode Linux, everything runs in the same addressing space.
- Hardware feature available on Skylake-SP, Zen 3 and later, available on client processors since Comet Lake.
- Memory protection keys are to provide isolation between different processes, and between processes and the kernel. Despite all of them being in the same address space, MPK allows to have isolation between components*.

*outside of side-channels

ARM64

- Currently shipping for Azure SQL Edge.
- The LibOS (and the other SFPs too) are identical between x86_64 and Arm 64-bit. Executables are statically translated with sbtrans.
- The translations are stored in lib/sbt in the ELF format. No JIT fallback is present in the currently shipped product.

DRTL: RUNTIME LIBRARY

AN ABSTRACTION LAYER

DRTL API (INCOMPLETE)

Streams:
DrtlOpenStream
DrtlReadStreamSync
DrtlWriteStreamSync
Threads:
DrtlGetThreadInfo
DrtlGetCurrentThreadId
DrtlThreadExit
DrtlDelayCurrentThreadExecution
DrtlGetThreadHostParameter
Other:
DrtlObjectClose
DrtlSetPrivateData
DrtlRunningInKernelMode
DrtlGetSystemProcessObject
DrtlIsDebuggerPresent
DrtlIsCriticalSectionLocked
DrtlRefreshDebuggerPresence
DrtlCreateSystemProcessInfo
DrtlPrepareForHostCall

DrtlpAllocate
DrtlProtectVirtualMemory
DrtlDbgPrint
DrtlGetIsHostCpuVirtual
DrtlIsInEnclave
DrtlGetHostEnclaveType
DrtlpTranslateXmmRegister

DrtlpInitialize
DrtlpCanContinueWithInitialization
DrtlpBindToDkPal
DrtlpSetStartModuleLocation
DrtlpInitializeTeb
DrtlpInitializeTebRegisters
DrtlBootLibraryOs
DrtlpConfigureLibraryOs
DrtlDbgPrintEarlyBoot

DrtlpMonitorSignals
DrtlpHandleUserSharedDataAcess
DrtlpHandleException
DrtlpHandleIllegalInstruction

g_DrtlHeaps (global variable)
g_DrtlNumberHeaps (global variable)

DRAWBRIDGE RUNTIME LIBRARY

- Showed in some MS documents & videos, a Channel 9 video is what has the most information about it.
- Not documented, and ntoskrnl.dll (now sqlpal.dll) symbols not available anywhere publicly. They're now available for some versions, but not all, and especially *not* the interesting builds.
- Consequences: unknown NTUM-internal API.

NT: THE LIBOS

USER-MODE KERNEL

LAUNCH-TIME OPTIONS

- PAL_ENABLE_BOOT_WITH_MINWIN boots in MinWin mode. This disables win32k.
- PAL_USE_LARGE_PAGE_DLLS enable large pages for DLLs.
- PAL_BOOT_WITH_MINIMAL_CONFIG
- PAL_EARLY_LOG_LEVEL controls the log level during PAL initialisation.

LAUNCH-TIME OPTIONS (PT 2)

- HTTP support saw a big change for Server 2019. It's now configurable via the PAL_HTTP option with the settings being None, PassThrough and Native.
- PAL_ENABLE_PAGE_FAULT_POLICY
- PAL_UNC_MAPPING
- PAL_TIMER_QUANTUM_MILLISECONDS
- PAL_STOP_ON_GUEST_PROCESS_FAULT
- PAL_SOS_TRACE_FLAGS

LAUNCH-TIME OPTIONS (PT 3)

- PAL_PROGRAM_INFO to get info about the current build of the PAL. This also loads and prints the version number of the SFPs.
- Bug: in an unpacked configuration, the PAL crashes while trying to get the version number of the SFPs.
- And then there's more…

THE DIFFERENT SFP ARCHIVES

- system.sfp contains the NT user-mode kernel and Drawbridge-specific drivers. It also includes the CsrLoader and AppLoader bootstrap processes.
- system.common.sfp contains the components shared with regular Windows, with .dbpatch files where required. dbpatch files allow the executables to stay unmodified.
- system.netfx.sfp is .NET (non-core).
- system.security.sfp covers some cryptography DLLs, with an oddball implementation with odd sections such as gsspr.
- system.certificates.sfp has the certificates for the OS root store.

THE REGISTRY ON DRAWBRIDGE

- In the manifest, it's possible to define persistent hives as a part of the registry, that are mounted to a given location.
- For non-persistent parts of the registry, multiple hives exist for different component, and the file format is shared with desktop Windows. All those hives are mounted to the very same registry root.
- The text versions of the registries are present under the .dbreg extension inside of the SFPs.

RUNNING DRIVERS

- Afd (with AfdWin if the host OS is NT), afdwsk
- DevApi
- Http (minioborrowed_rs2), HttpPassThrough*
- SbsExt
- dxgkrnl
- fwpkclnt, ndis, netio
- Npfs
- Videodrv
- Cng, ksecdd, ksecpkg
- Hidparse
- Msrpc
- werkernel, wpprecorder
- NullDD
- … and win32k, which is disabled in a MinWin configuration.

RUNNING PROCESSES

- AppLoader
- CsrLoader
- Service host (telemetry + cryptography + RPC…)
- LSA
- DTC
- …and the app
- The UM kernel spawns CsrLoader then AppLoader, which handles the initialisation of the system.

SOME NOTES

- System information class 0x1388 allows to determine if a given task is running within a Drawbridge LibOS.
- The address space is shared across multiple processes in practice.
- The Console* APIs, which don't currently have an implementation on the PAL side, allow GUI apps to run on Drawbridge.
- Tip: you can unpack the SFP and run a LibOS just fine. The name of the directory has to end with .sfp.
- DkDll, DNS and system.security DLLs have .gsspr and .gssep sections, it's an SS guard implemented in MSVC as a non-public flag.

THE KERNEL ITSELF

- The NTUM is named sqlpal.dll and is identified as build 14388 in the file information, along with Drawbridge-specific drivers.
- ntdll and everything in packages other than system.sfp is built for build 17134.
- SQL Server 2017 used build 15063 as a base, and a 15063 build using the new syscall ABI exists in early SQL Server 2019 previews.
- Not only NT calls are exposed but SOSv2 specific ones too.

DKDLL SYSCALLS

- There’s only a very reduced number of them which are:
- NtGenerateRandomData: 0x3000
- NtReportUnimplemented: 0x3001
- NtGetExternalPid: 0x3002
- NtSwitchTeb: 0x3003
- NtServiceSwitchTeb: 0x3004

DNS

- DNS is implemented not with passing information from /etc/hosts and /etc/resolv.conf from the host OS but via a dnsapi implementation.
- \\??\\StreamUri\dns is the way used under the hood by dnsapi to get the IP from a hostname.

PATCH FILES

- Magic at the beginning of the file: dbpatch
- Implemented as a LibOS feature.
- Gone in SQL Server 2019 RTM. .dbpatch files are now embedded in the executable for win32u.dll, ntdll.dll and dkdll.dll, without mentions of it anywhere else except the kernel.

SECURITY SHIM FORWARDER LIBRARIES

- Instead of using native Windows security libraries directly, security shim forwarder libraries are present.
- Present DLLs are: advapi32, *bcrypt, ncrypt*, rpcrt4, secur32.
- Forwards to _NT variants for functions that aren't part of the modified bits. (looking at advapi32 and rpcrt4 for example, which aren't all crypto stuff)
- You can look at securityapi.dll in system.sfp.
- The DLL names in italics aren't forwarders, at least in 2019 RTM.

DEMO

RUNNING WINDOWS APPLICATIONS FROM THE OUTSIDE ON DRAWBRIDGE

WHAT THIS WON'T DO

- On Arm, this won't work. Your new executable doesn't have a corresponding sbtrans cache, so you won't get anywhere.
- GUIs, those were not tested as they aren't part of the scope of this effort. Your mileage might vary.
- If your executable is 32-bit, this won't run.
- If your executable isn't relocatable, your mileage might vary.
- Some other limitations. This list isn't exhaustive.

DEMO

- PowerShell Core running on Drawbridge using the Windows binary.
- No stdin flushing support is present in the console implementation here.

```
PS /home/laptop2/dev/drawbridge/makecustom> ./pwsh-drawbridge
PowerShell 6.1.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS C:\Windows\System32> dir

Oops, something went wrong.  Please report this bug with the details below.
Report on GitHub: https://github.com/lzybkr/PSReadLine/issues/new
----------------------------------------------------------------------
Last 1 Keys:
```
 i
```

Exception:
```
System.IO.IOException: Incorrect function
   at System.ConsolePal.set_CursorVisible(Boolean value)
   at Microsoft.PowerShell.PSConsoleReadLine.ReallyRender(RenderData renderData, String defaultColor)
   at Microsoft.PowerShell.PSConsoleReadLine.ForceRender()
   at Microsoft.PowerShell.PSConsoleReadLine.Insert(Char c)
   at Microsoft.PowerShell.PSConsoleReadLine.SelfInsert(Nullable`1 key, Object arg)
   at Microsoft.PowerShell.PSConsoleReadLine.ProcessOneKey(ConsoleKeyInfo key, Dictionary`2 dispatchTable, Boolean ignoreIfNoAction, Object arg)
   at Microsoft.PowerShell.PSConsoleReadLine.InputLoop()
   at Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics, CancellationToken cancellationToken)
```
----------------------------------------------------------------------
^C
An error has occurred that was not properly handled. Additional information is shown below. The PowerShell process will exit.

Unhandled Exception: System.Management.Automation.Host.HostException: The Win32 internal error "Incorrect function" 0x1 occurred while flushing the
 console input buffer. Contact Microsoft Customer Support Services. ---> System.ComponentModel.Win32Exception: Incorrect function
   --- End of inner exception stack trace ---
   at Microsoft.PowerShell.ConsoleControl.FlushConsoleInputBuffer(SafeFileHandle consoleHandle)
   at Microsoft.PowerShell.ConsoleHost.HandleBreak()
   at System.Threading.Thread.ThreadMain_ThreadStart()
   at System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state)
--- End of stack trace from previous location where exception was thrown ---
```

DEMO USING AN EARLIER SQL SERVER PREVIEW

```
testlab@testlab-Virtual-Machine:~/drawbridge/lab$ sudo ./palrun -Command "\$PSVe
rsionTable"
[sudo] password for testlab:
This is an evaluation version.  There are [116] days left in the evaluation peri
od.

Name                           Value
----                           -----
PSVersion                      6.2.0-preview.4
PSEdition                      Core
GitCommitId                    6.2.0-preview.4
OS                             Microsoft Windows 10.0.9200
Platform                       Win32NT
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0
```

Shows build 9200 instead of 15063, which is the used Windows version in this case.

SGX & VBS

ENCLAVES

SGX & VBS NOTES

- New API not present on the original Drawbridge.
- Minimal API set present, exposed through the newer Syscall mechanism only.
- The SGX support is implemented since a while on Linux. A 0xc00000bb error is returned if trying to use an unsupported enclave type. The PAL is statically linked to SGX AESM support code.

OTHER USES OF DRAWBRIDGE: BACKWARDS COMPATIBILITY

- Windows CE compatibility on Windows 10 IoT Core uses Drawbridge. Instead of NT running as the guest, Windows CE is used instead.
- That implementation has a more conventional PAL and monitor, with GUI support included. It's outside of the scope of this slide deck.

REFERENCES/LINKS

https://cloudblogs.microsoft.com/sqlserver/2016/12/16/sql-server-on-linux-how-introduction/
http://db.cs.duke.edu/courses/cps210/spring15/readings/VEE14-present601.pdf
https://www.microsoft.com/en-us/research/project/drawbridge/
https://channel9.msdn.com/shows//going+Deep//drawbridge-An-Experimental-Library-Operating-System
https://dl.acm.org/doi/abs/10.1145/2799647
https://www.usenix.org/conference/osdi14/technical-sessions/presentation/baumann
https://csrc.nist.gov/CSRC/media/Presentations/Haven-Shielding-Applications-from-an-Untrusted-Cl/images-media/day2_trusted-computing_1100-1150.pdf
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/osdi2014-haven.pdf
https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/asplos2011-drawbridge.pdf

QUESTIONS?

OR NOT… YOU CAN SEND THEM TO

NEVER_RELEASED@THREEDOTS.OVH