



iDRACKAR

integrated Dell Remote Access Controller's Kind Approach to the RAM

Nicolas looss

SSTIC 2019



Plan

- 1 Introduction
- 2 Approche logicielle
- 3 Composants matériels spécifiques : CPLD, PBI, etc.
- 4 Le chaînon manquant



À quoi sert un BMC / iDRAC ?

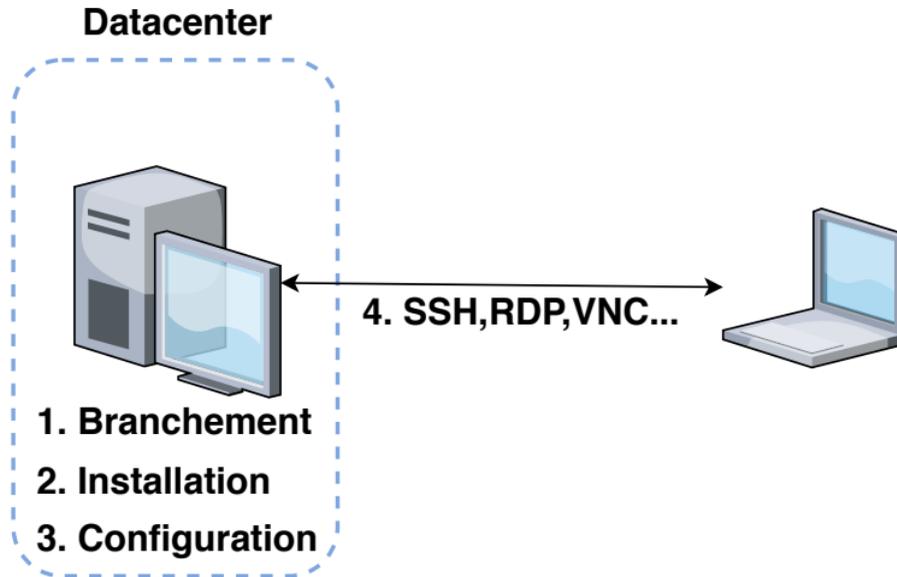


FIGURE 1 – Mise en place d'un serveur



À quoi sert un BMC / iDRAC ?

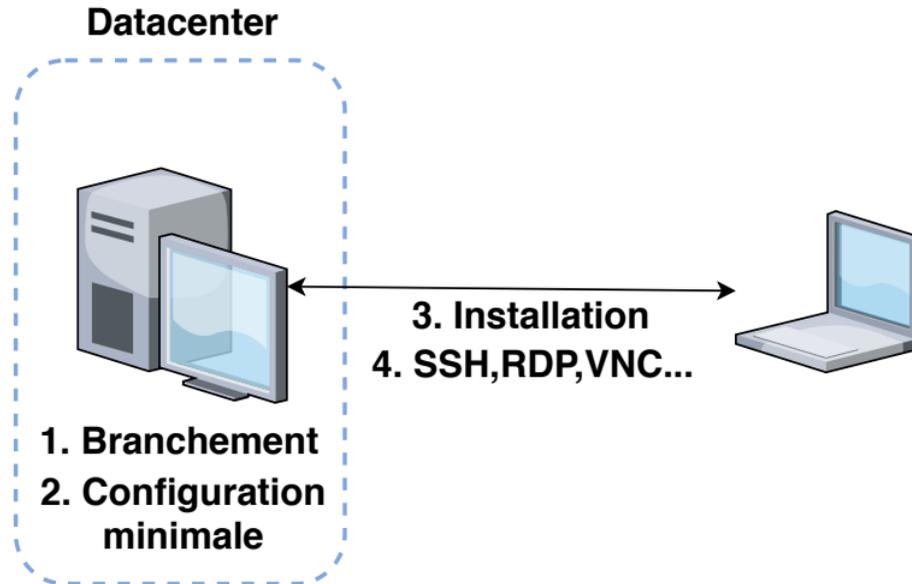


FIGURE 2 – Installation à distance grâce au BMC



L'iDRAC, une implémentation de BMC



FIGURE 3 – Ports d'un serveur Dell PowerEdge R730



Réseau avec BMC

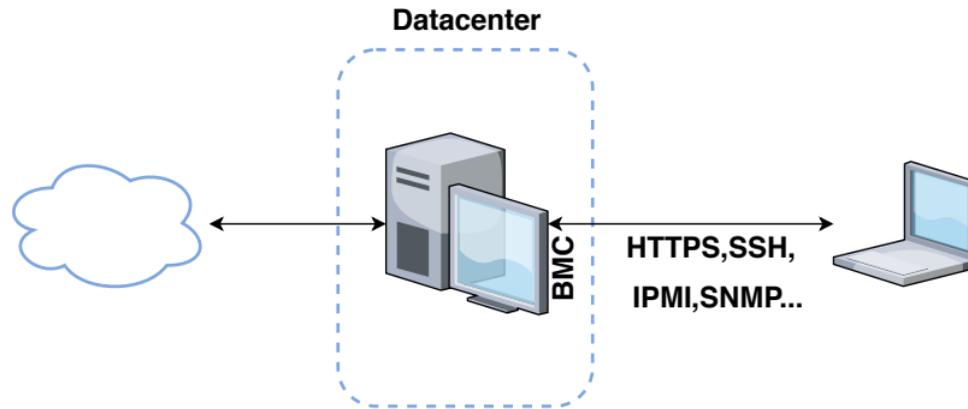


FIGURE 4 – En pratique, interfaces réseau séparées



Acronymes

- ▶ BMC : Baseboard Management Controller
- ▶ OOB : Out-Of-Band (management)

Implémentations :

- ▶ iLO : BMC de HP
- ▶ iDRAC : BMC de Dell
- ▶ iLOM : BMC d'Oracle
- ▶ IMM : BMC de Lenovo
- ▶ AMT/ME/CSME/... : BMC d'Intel
- ▶ OpenBMC : implémentation open-source
- ▶ ...



Versions de l'iDRAC

Année	Version	Génération de serveur Dell
1999	DRAC II	
2002	DRAC III	
2005	DRAC IV	8 ^e
2006	DRAC 5	9 ^e
2008	iDRAC 6	
2012	iDRAC 7	12 ^e
2014	iDRAC 8	13 ^e
2017	iDRAC 9	14 ^e



Versions de l'iDRAC

Année	Version	Génération de serveur Dell
1999	DRAC II	
2002	DRAC III	
2005	DRAC IV	8 ^e
2006	DRAC 5	9 ^e
2008	iDRAC 6	
2012	iDRAC 7	12 ^e
2014	iDRAC 8	13^e
2017	iDRAC 9	14 ^e



Serveur Web embarqué

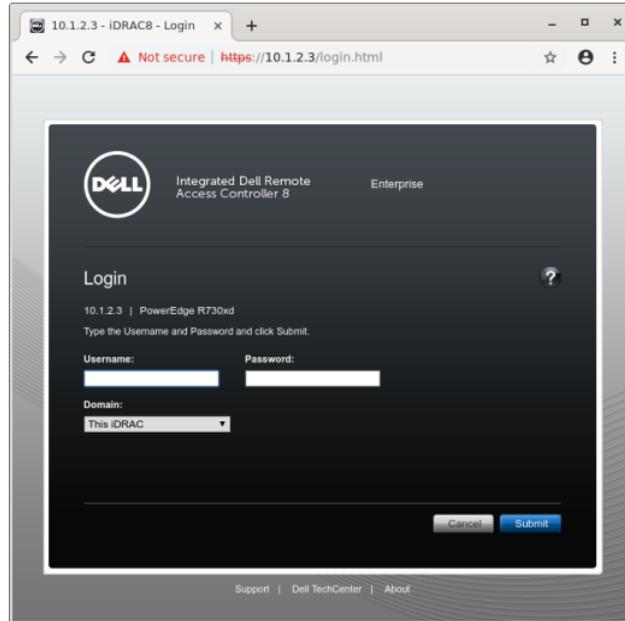


FIGURE 5 – Écran de connexion d'un iDRAC 8



Serveur Web embarqué

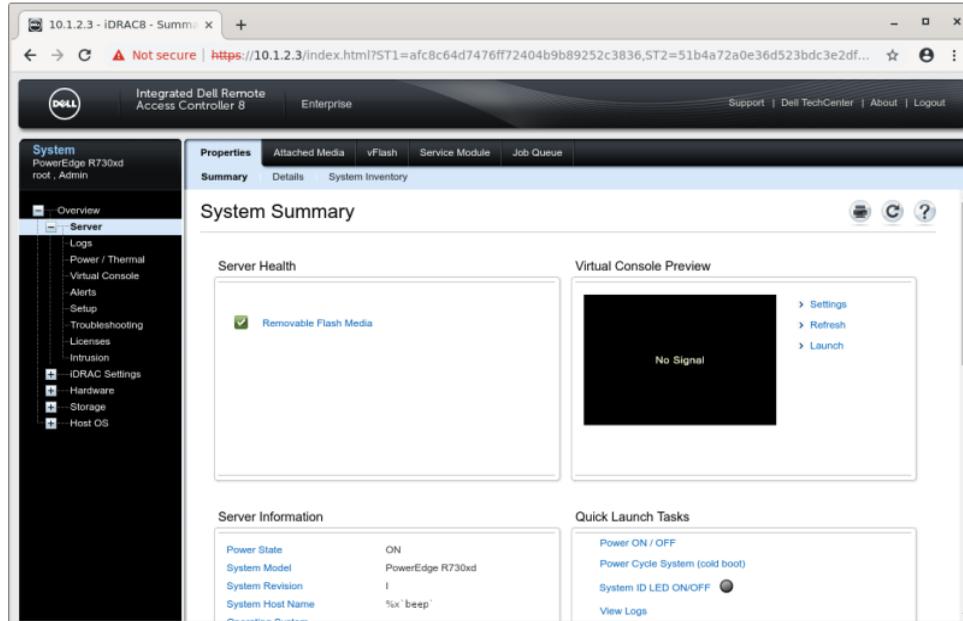


FIGURE 6 – Page Web principale de l'iDRAC 8



Vulnérabilité

CVE-2018-1207 : injection de code arbitraire sur iDRAC 7 ou 8 version \leq 2.52.52.52

- ▶ depuis le serveur Web
- ▶ avant authentification

cf. Black Hat USA 2018 : *The Unbearable Lightness of BMC's*



Vulnérabilité

CVE-2018-1207 : injection de code arbitraire sur iDRAC 7 ou 8 version \leq 2.52.52.52

- ▶ depuis le serveur Web
- ▶ avant authentification

cf. Black Hat USA 2018 : *The Unbearable Lightness of BMC's*

Un attaquant peut directement :

- ▶ voir le contenu de l'écran physiquement connecté ;
- ▶ utiliser le clavier/souris virtuel et le *Virtual Media* ;
- ▶ modifier la configuration du démarrage ;
- ▶ éteindre/redémarrer le serveur ;
- ▶ communiquer en utilisant l'interface réseau de l'iDRAC ;
- ▶ etc.

Position similaire à un accès physique.



Le compte manquant

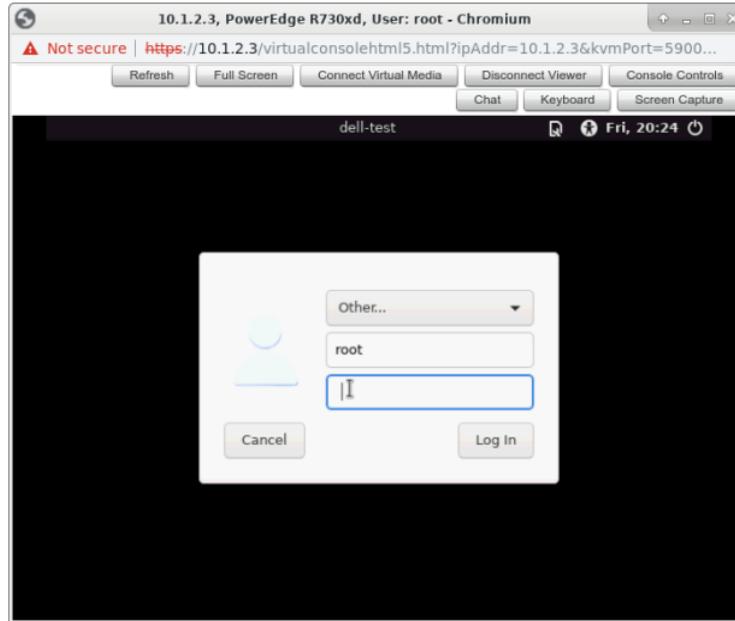


FIGURE 7 – Console déportée de l'iDRAC avec une demande d'authentification du système principal



Est-ce qu'un attaquant peut accéder au contenu de la mémoire principale ?

Cela permettrait :

- ▶ d'obtenir des clés de chiffrement du disque dur ;
- ▶ d'injecter du code dans le système principal ;
- ▶ d'accéder aux autres interfaces réseau du serveur ;
- ▶ etc.

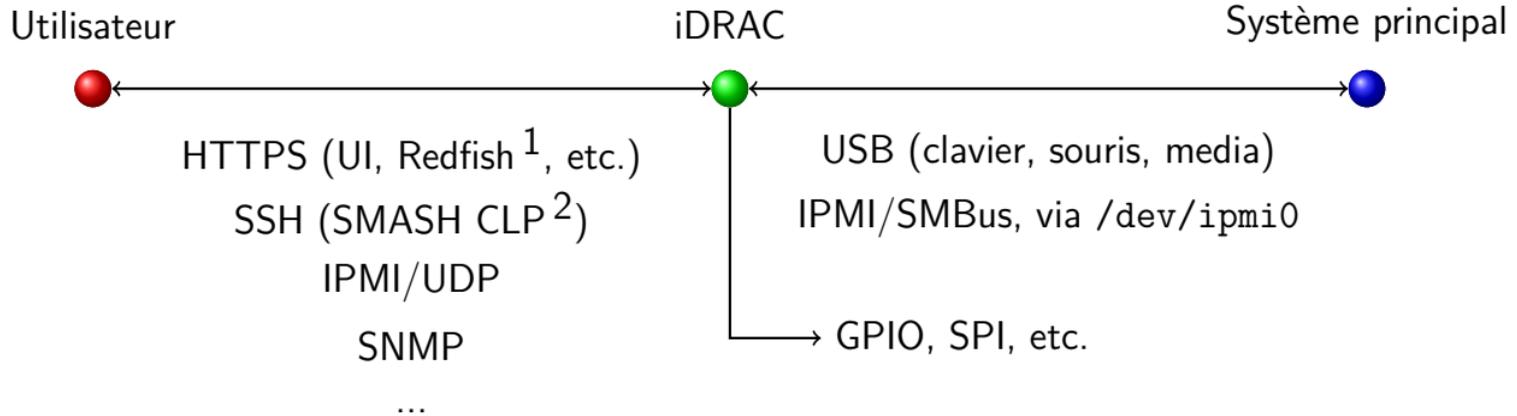


Plan

- 1 Introduction
- 2 Approche logicielle
- 3 Composants matériels spécifiques : CPLD, PBI, etc.
- 4 Le chaînon manquant



Communications de l'iDRAC

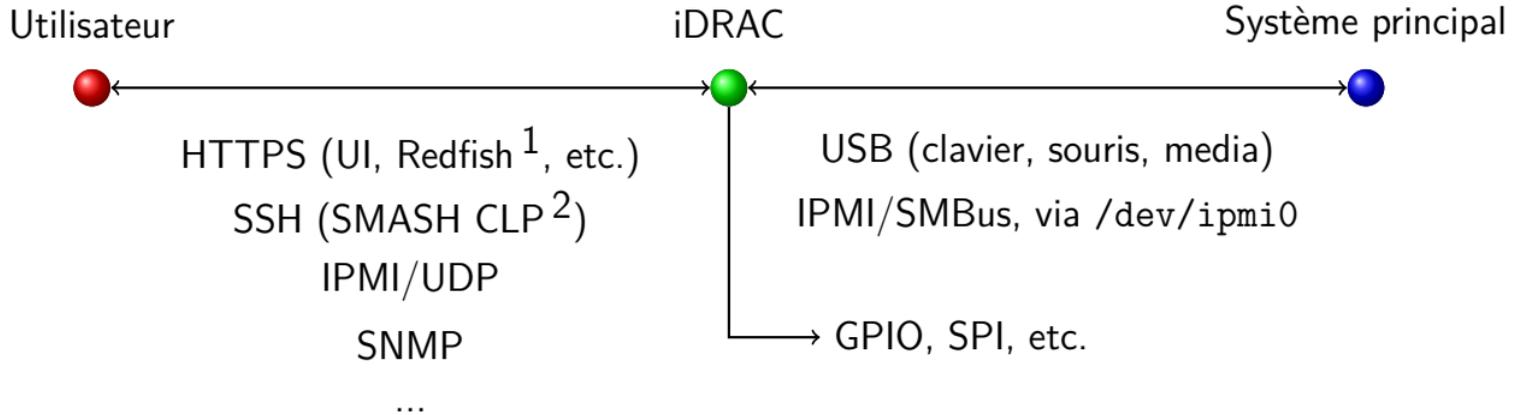


1. Interface JSON:API accessible sur /redfish/v1

2. Systems Management Architecture for Server Hardware - Command Line Protocol



Communications de l'iDRAC



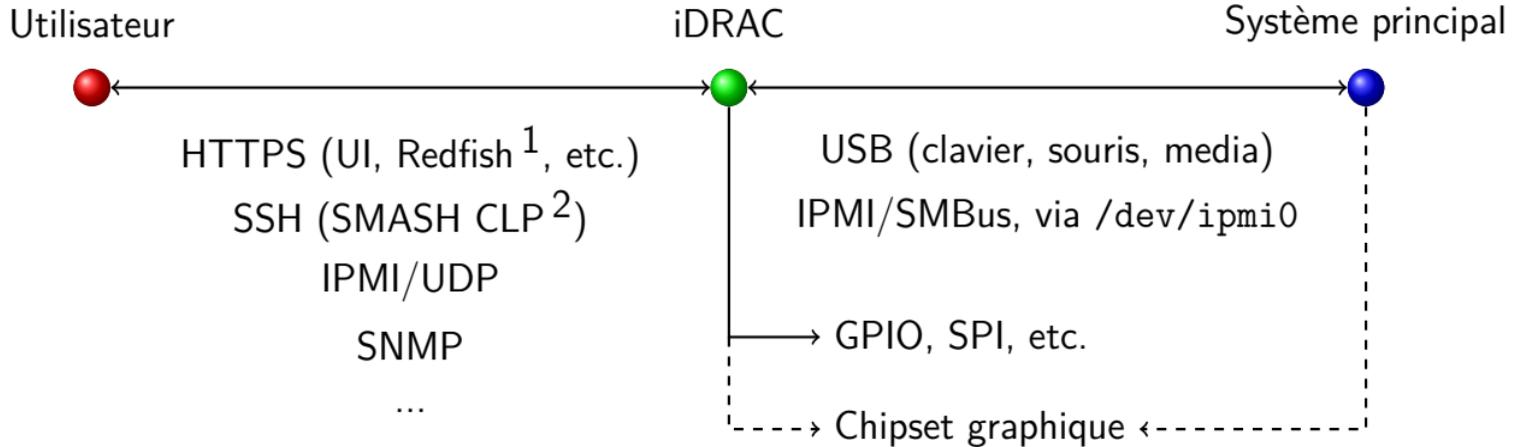
Et l'écran déporté? Probablement partagé

Et le PCIe?

1. Interface JSON:API accessible sur /redfish/v1
2. Systems Management Architecture for Server Hardware - Command Line Protocol



Communications de l'iDRAC



Et l'écran déporté? Probablement partagé

Et le PCIe?

1. Interface JSON:API accessible sur /redfish/v1
2. Systems Management Architecture for Server Hardware - Command Line Protocol



Où trouver de l'information ?

- ▶ Documentation sur Internet
- ▶ Mises à jour : archives signées
- ▶ Shell obtenu par CVE-2018-1207
- ▶ Mais aussi...



Les sources ouvertes

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `https://opensource.dell.com/releases/idrac8/`. The page title is "Index of /releases/idrac8". The content is a directory listing table with columns for Name, Last modified, Size, and Description.

Name	Last modified	Size	Description
Parent Directory	-	-	-
License_manifest_2.20.20.20.txt	2018-09-25 06:32	18K	
License_manifest_2.30.30.30.txt	2018-09-25 06:33	18K	
License_manifest_2.40.40.40.txt	2018-09-25 06:33	17K	
License_manifest_2.50.50.50.txt	2018-09-25 06:33	17K	
License_manifest_2.60.60.60.txt	2018-09-26 05:14	17K	
SHA256SUMS	2018-09-26 10:23	1.3K	
iDRAC_opensource_2.20.20.20.tar.gz	2016-01-22 12:10	1.0G	
iDRAC_opensource_2.30.30.30.tar.gz	2016-08-22 11:21	1.0G	
iDRAC_opensource_2.40.40.40.tar.gz	2016-11-18 11:04	1.0G	
iDRAC_opensource_2.50.50.50.tar.gz	2018-01-15 23:31	1.1G	
iDRAC_opensource_2.60.60.60.tar.gz	2018-07-03 04:51	1.1G	
opensource_2.00.00.iso	2015-06-22 20:58	2.9G	
opensource_2.05.05.iso	2015-06-22 21:01	2.4G	
opensource_2.10.10.iso	2015-06-22 21:02	2.4G	

FIGURE 8 – Code source (et programmes compilés) sur `https://opensource.dell.com/`



Les sources ouvertes

Pourquoi ?

- ▶ Distribution Linux modifiée (modules Dell sous licence GPL)
- ▶ U-Boot, OpenSSH, systemd, etc.



Les sources ouvertes

Pourquoi ?

- ▶ Distribution Linux modifiée (modules Dell sous licence GPL)
- ▶ U-Boot, OpenSSH, systemd, etc.

```
[SH7757 /flash/data0/home/root]$ id
uid=0(root) gid=0(root) groups=0(root)
```

```
[SH7757 /flash/data0/home/root]$ uname -a
Linux MpCOZ1Z 3.4.11 #1 Thu Aug 18 13:03:21 CDT 2016 sh4a GNU/Linux
```

```
[SH7757 /flash/data0/home/root]$ cat /etc/issue
Poky 8.0 (Yocto Project 1.3 Reference Distro) 1.3 \n \l
```

```
[SH7757 /flash/data0/home/root]$ ls -l /sbin/init
lrwxrwxrwx 1 root 0 20 Aug 18 2016 /sbin/init -> /lib/systemd/systemd
```



Qu'y a-t-il à l'intérieur du serveur ?

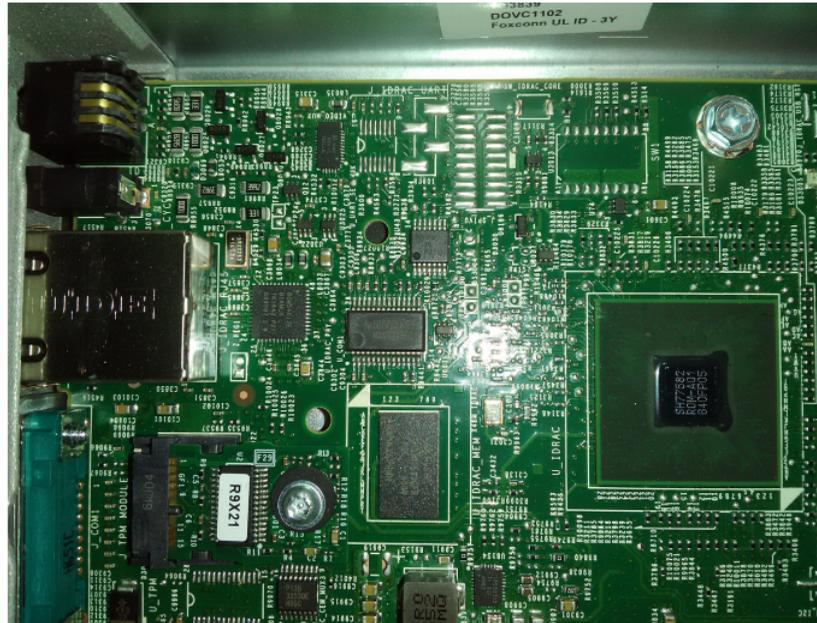
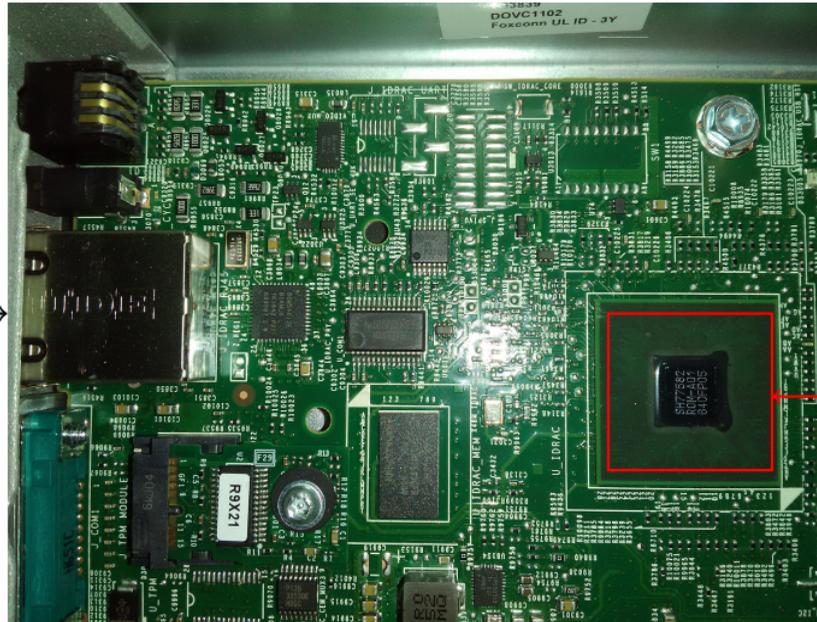


FIGURE 9 – CPU de l'iDRAC 8



Qu'y a-t-il à l'intérieur du serveur ?

Ethernet
« iDRAC »



SH77582
(CPU SuperH4)

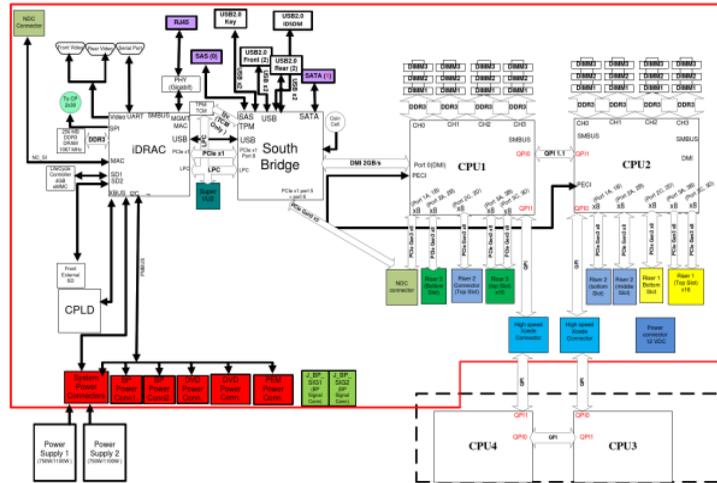
FIGURE 9 – CPU de l'iDRAC 8



Dell PowerEdge R820 system board block diagram

Appendix C. System board block diagram

Figure 14. R820 system board block diagram



54 PowerEdge R820 Technical Guide



FIGURE 10 – <https://www.manualslib.com/manual/624251/Dell-Poweredge-R820.html?page=54>



Que voit le système d'exploitation principal ?

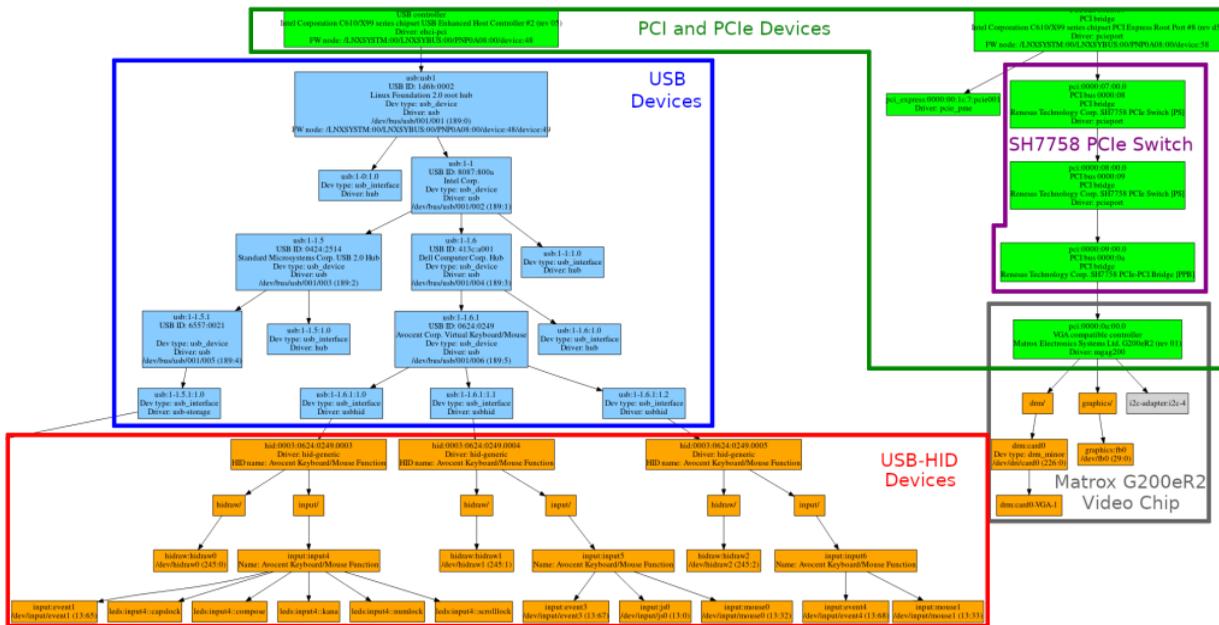


FIGURE 13 – sortie de <https://github.com/fishilico/home-files/blob/master/bin/graph-hw>



Arborescence PCIe

BDF = Bus, Device, Function (triplet identifiant un périphérique PCI)

BDF	Vendor ID	Device ID	Vendor	Device name
00:1c.7	8086	8d1e	Intel	PCI Express Root Port #8
07:00.0	1912	001d	Renesas	SH7758 PCIe Switch [PS]
08:00.0	1912	001d	Renesas	SH7758 PCIe Switch [PS]
09:00.0	1912	001a	Renesas	SH7758 PCIe-PCI Bridge [PPB]
0a:00.0	102b	0534	Matrox	G200eR2

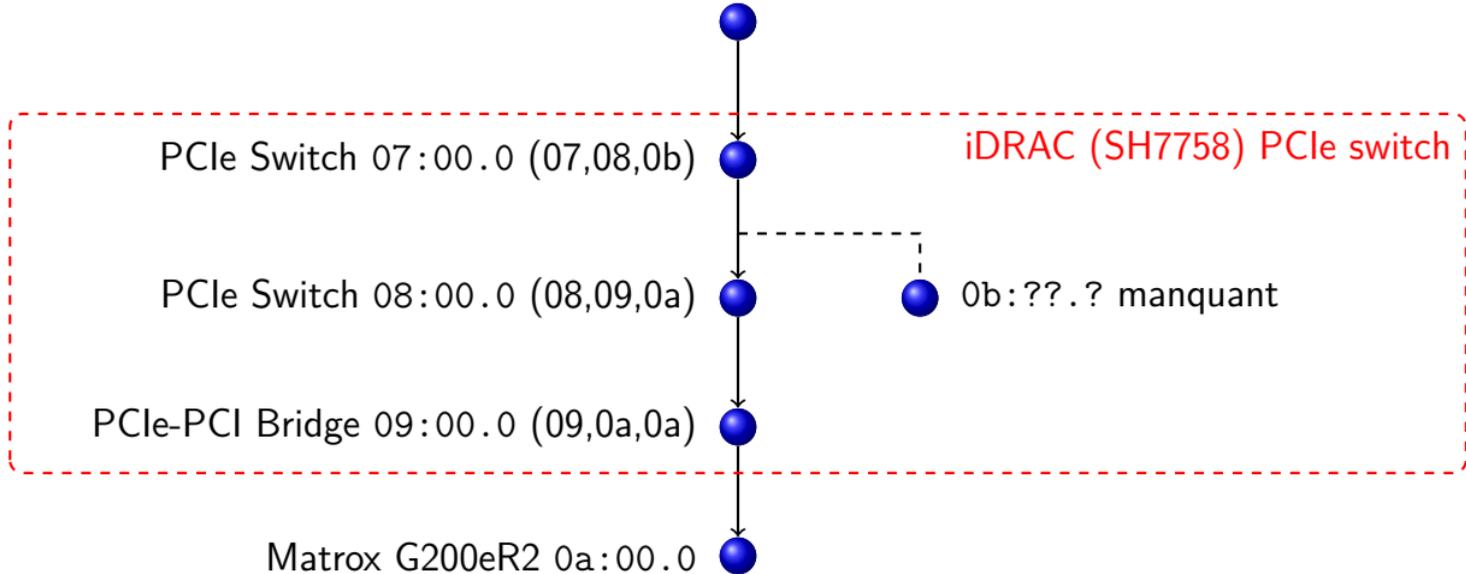
```
$ lspci -t
```

```
+1c.7-[07-0b]----00.0-[08-0b]----00.0-[09-0a]----00.0-[0a]----00.0
```



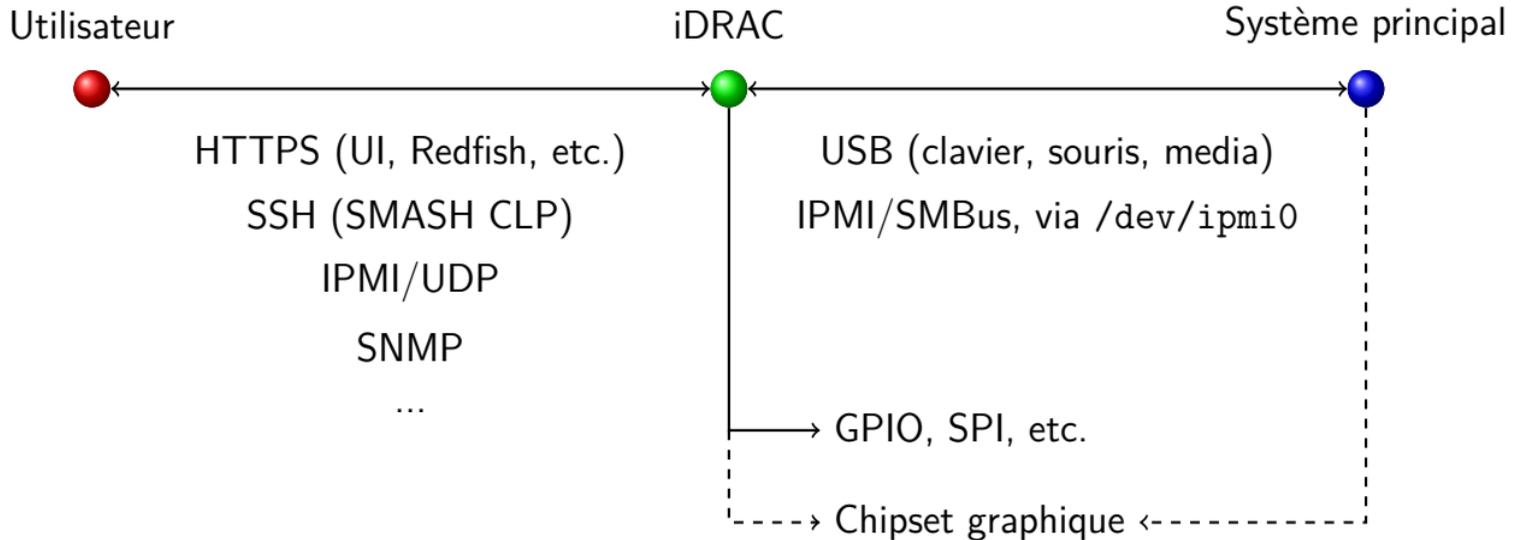
Arborescence PCIe

PCIe Root Port #8 00:1c.7 (Pri=00,Sec=07,Sub=0b)



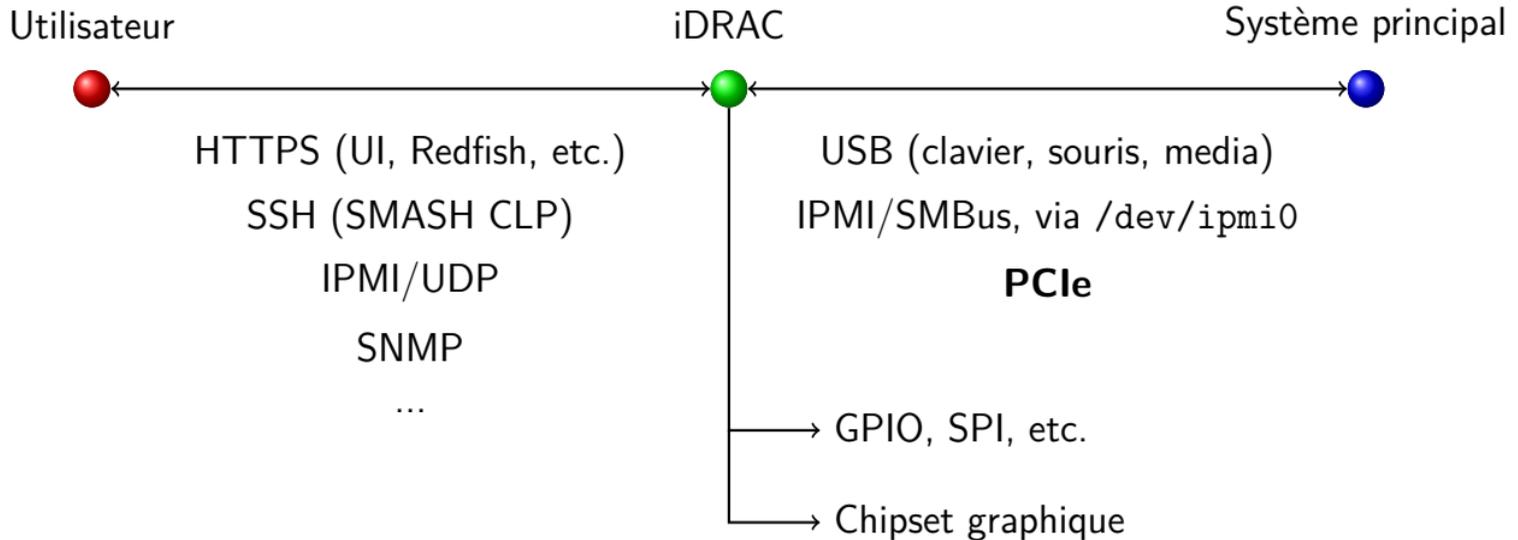


Communications de l'iDRAC - reprise





Communications de l'iDRAC - reprise





Est-ce qu'un attaquant peut accéder au contenu de la mémoire principale ?



Est-ce qu'un attaquant peut accéder au contenu de la mémoire principale ?

Un attaquant peut utiliser un module *USB-Gadget* pour connecter un périphérique USB virtuel.

- ▶ utilisé pour clavier, souris, *Virtual Media* ;
- ▶ utilisé pour une interface Ethernet-USB interne :

```
racadm set iDRAC.OS-BMC.AdminState Enabled
```

Il peut accéder au SMBus et à d'autres bus à faible débit.

Mais aussi au bus PCIe, car l'iDRAC est en coupure avec la carte graphique.

Comment ?



Plan

- 1 Introduction
- 2 Approche logicielle
- 3 Composants matériels spécifiques : CPLD, PBI, etc.**
- 4 Le chaînon manquant



Le CPLD

- ▶ CPLD = Complex Programmable Logic Device (circuit intégré reprogrammable)
- ▶ Comme un FPGA, mais moins complexe et avec une mémoire persistante optionnelle.

Est-ce que le CPLD implémente du PCIe ?



U_CPLD à côté de U_IDRAC



FIGURE 14 – CPLD à côté du CPU de l'iDRAC



U_CPLD à côté de U_IDRAC

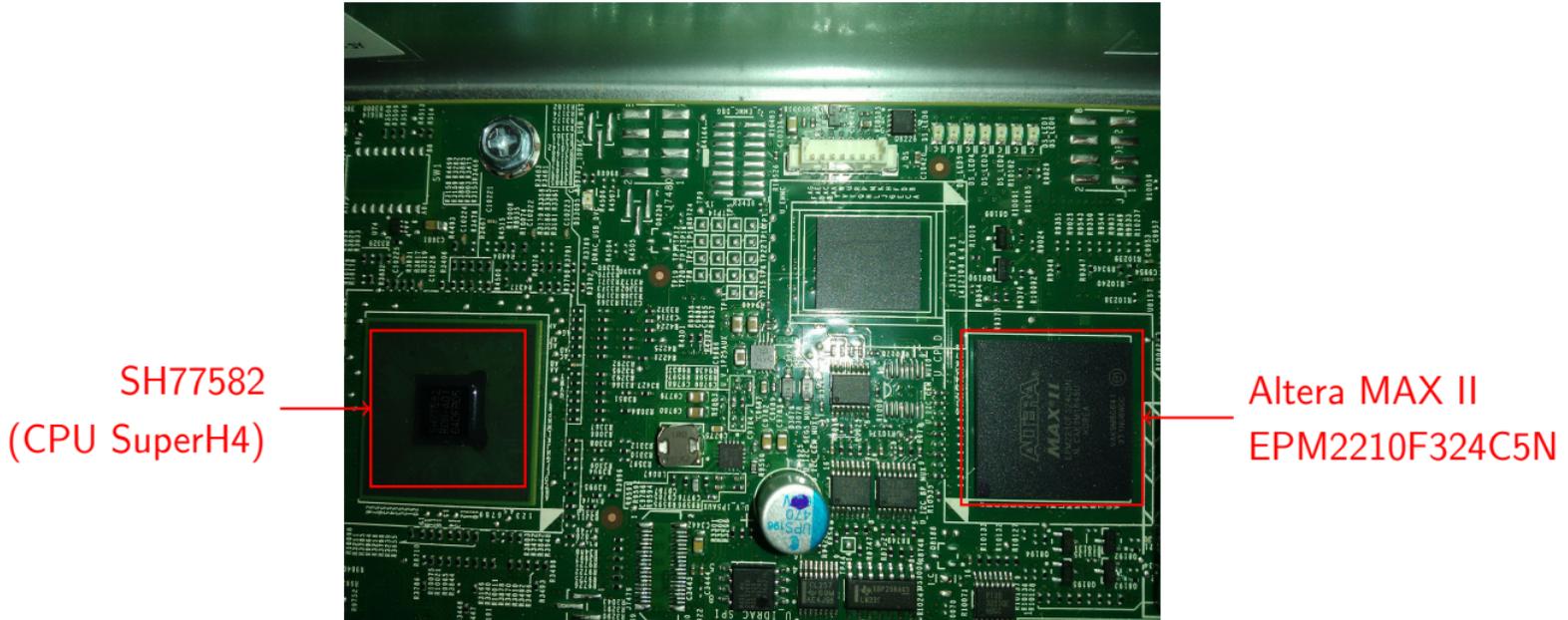


FIGURE 14 – CPLD à côté du CPU de l'iDRAC



Utilisation du CPLD

- ▶ Stockage de quelques informations :

```
$ cat bin/throttle.sh
```

```
[...]
```

```
#-----
```

```
# Get Planer Type id
```

```
#-----
```

```
PLANTYPE1=0x`MemAccess -rb 0x14000003|tail -n +4|head -n 1|cut -f 3 -d ' '`
```

```
PLANTYPE2=0x`MemAccess -rb 0x14000004|tail -n +4|head -n 1|cut -f 3 -d ' '`
```

- ▶ GPIO pour la connexion des périphériques USB virtuels.

⇒ Pas de PCIe :(



Utilisation du CPLD

- ▶ Stockage de quelques informations :

```
$ cat bin/throttle.sh
```

```
[...]
```

```
#-----
```

```
# Get Planer Type id
```

```
#-----
```

```
PLANTYPE1=0x`MemAccess -rb 0x14000003|tail -n +4|head -n 1|cut -f 3 -d ' '`
```

```
PLANTYPE2=0x`MemAccess -rb 0x14000004|tail -n +4|head -n 1|cut -f 3 -d ' '`
```

- ▶ GPIO pour la connexion des périphériques USB virtuels.

⇒ Pas de PCIe :(

MemAccess et MemAccess2 : accès à la mémoire physique *de l'iDRAC*



Le PBI

Sur <https://certification.ubuntu.com/>

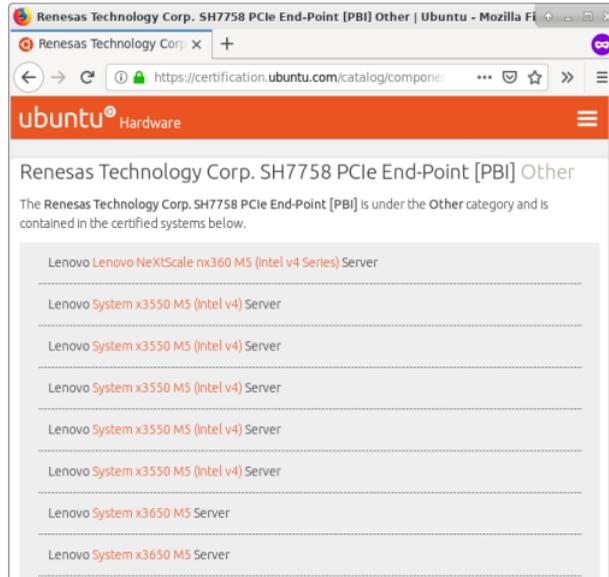


FIGURE 15 – Renesas Technology Corp. SH7758 PCIe End-Point [PBI] (1912:001b)



grep "PBI"

```
externalsrc/linux-drivers/pbi_driver/sh_pbi.c (pour /dev/sh_pbi) :
#define PBI_MBOX_SIZE          0x1000
#define PBI_MBOX_REGS          0xffca0000
#define PBI_SMEM_SIZE          0x1000
#define PBI_SMEM_START         0xffcaa000
/* ... */
static long
sh_pbi_mbox_ioctl(struct file *file, unsigned int cmd, unsigned long arg)
    switch( cmd ) {
        case DELL_PBI_PRINT_CONFIG:
            // Print PCI configuration space
            printk("\n%s::%s()PCI_CONFIG\n",DRIVER_NAME,__FUNCTION__);
```



Commande pbittest dans l'iDRAC 8

```
Usage: pbittest command [parameter1] [parameter2]
ex: pbittest 1 <button> <direction> (send button message)
      button 21=select 22=right 23=left
      direction 0=release 1=press
ex: pbittest 2 (send response info message)
ex: pbittest 3 (send reset message)
ex: pbittest 4 (read message)
ex: pbittest 5 (poll message)
ex: pbittest 6 (msgin poll message)
ex: pbittest 7 <rate> <iterations> (button stress test)
      rate = key events per seconds, iterations is total times to run
ex: pbittest 8 (dumps PBI config space
      use dmesg to display output)
ex: pbittest 9 (PBI_MAILBOX_RESET)
```



PBI dans la RAM de l'iDRAC 8

MemAccess 0xffca0000 + lspci :

Renesas Technology Corp. SH7758 PCIe End-Point [PBI] [1912:001b]

Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR- FastB2B- DisINTx-

Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort-
<TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0

Interrupt: pin A routed to IRQ 255

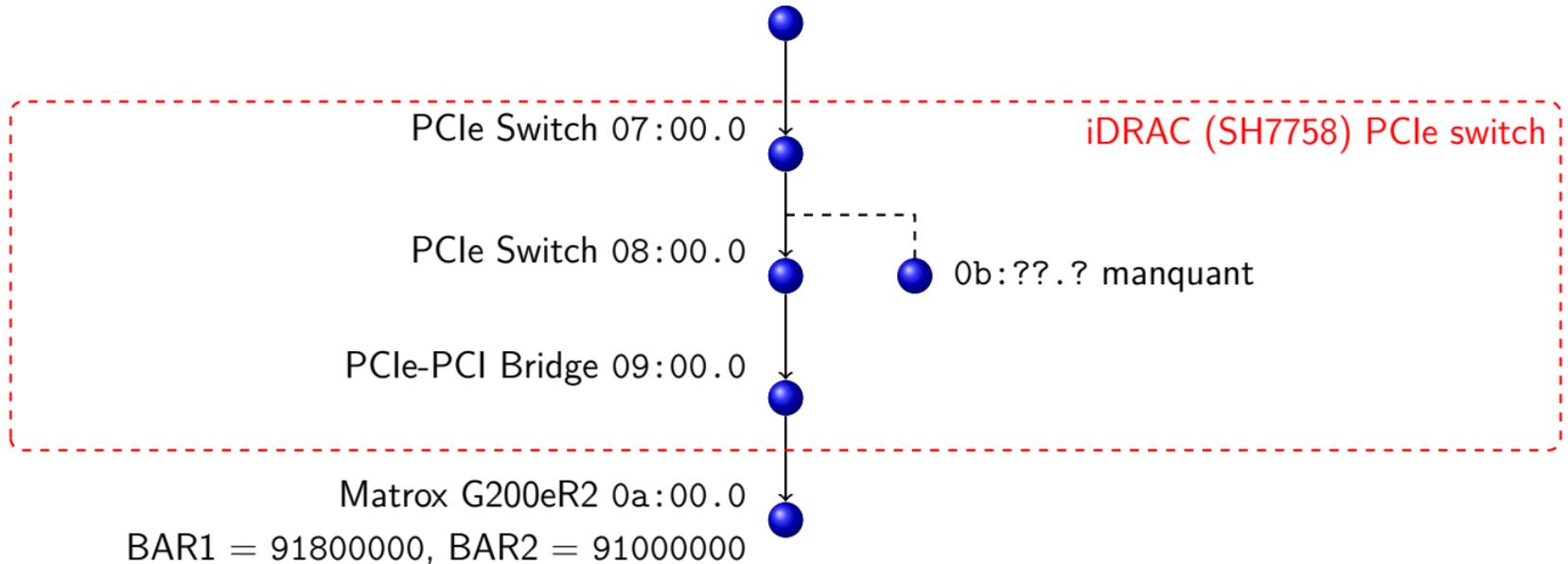
Region 0: Memory at 91901000 (32-bit, non-prefetchable)

Region 1: Memory at 91900000 (32-bit, non-prefetchable)



Arborescence PCIe avec les adresses

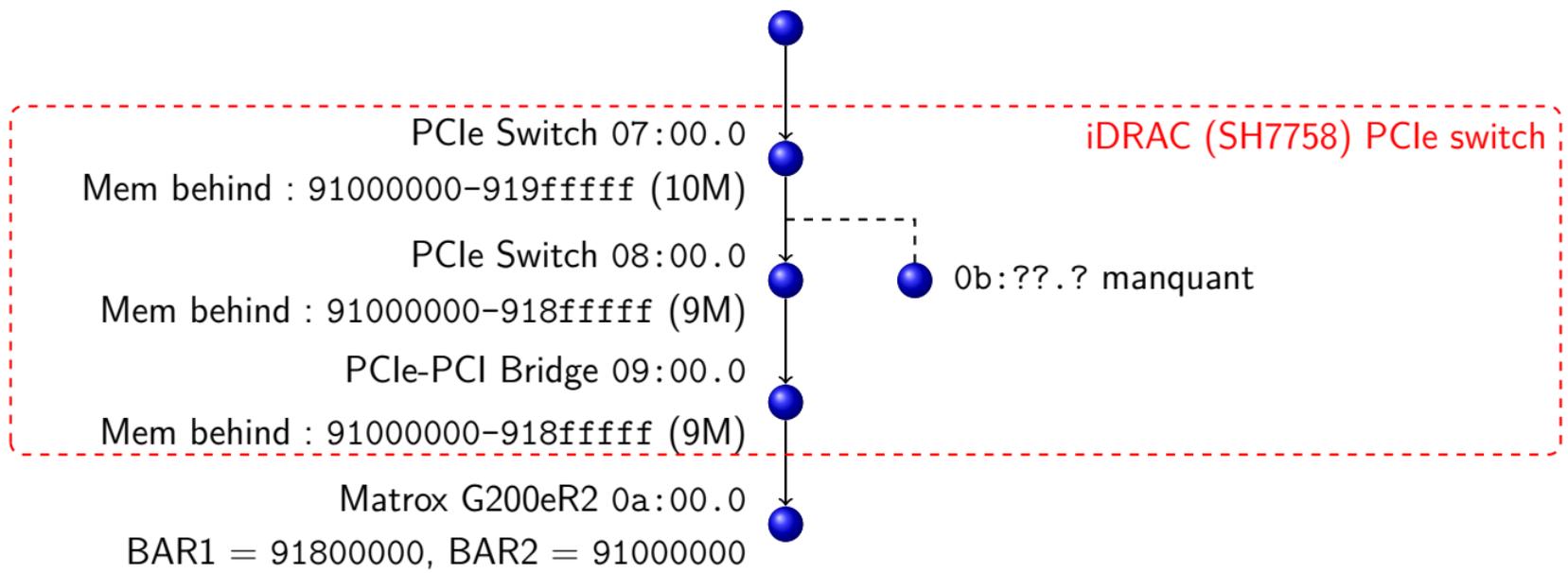
PCIe Root Port #8 00:1c.7 (Memory behind bridge : 91000000-919fffff (10M))





Arborescence PCIe avec les adresses

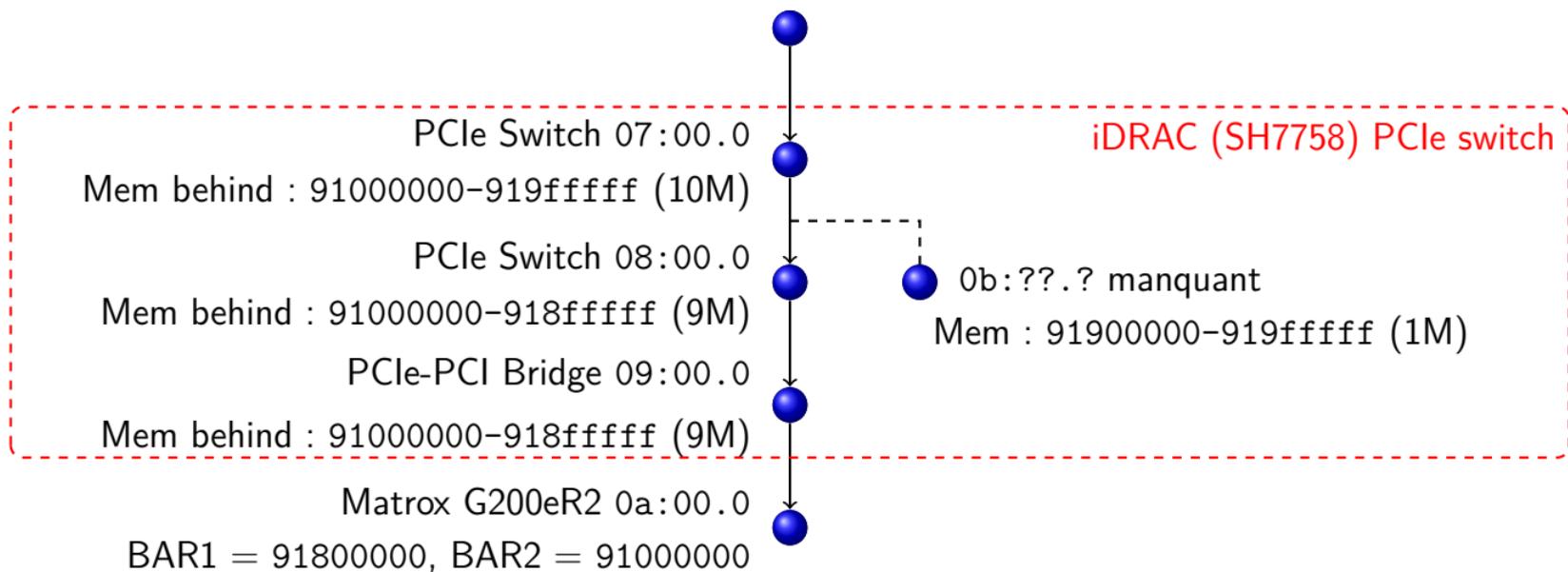
PCIe Root Port #8 00:1c.7 (Memory behind bridge : 91000000-919fffff (10M))





Arborescence PCIe avec les adresses

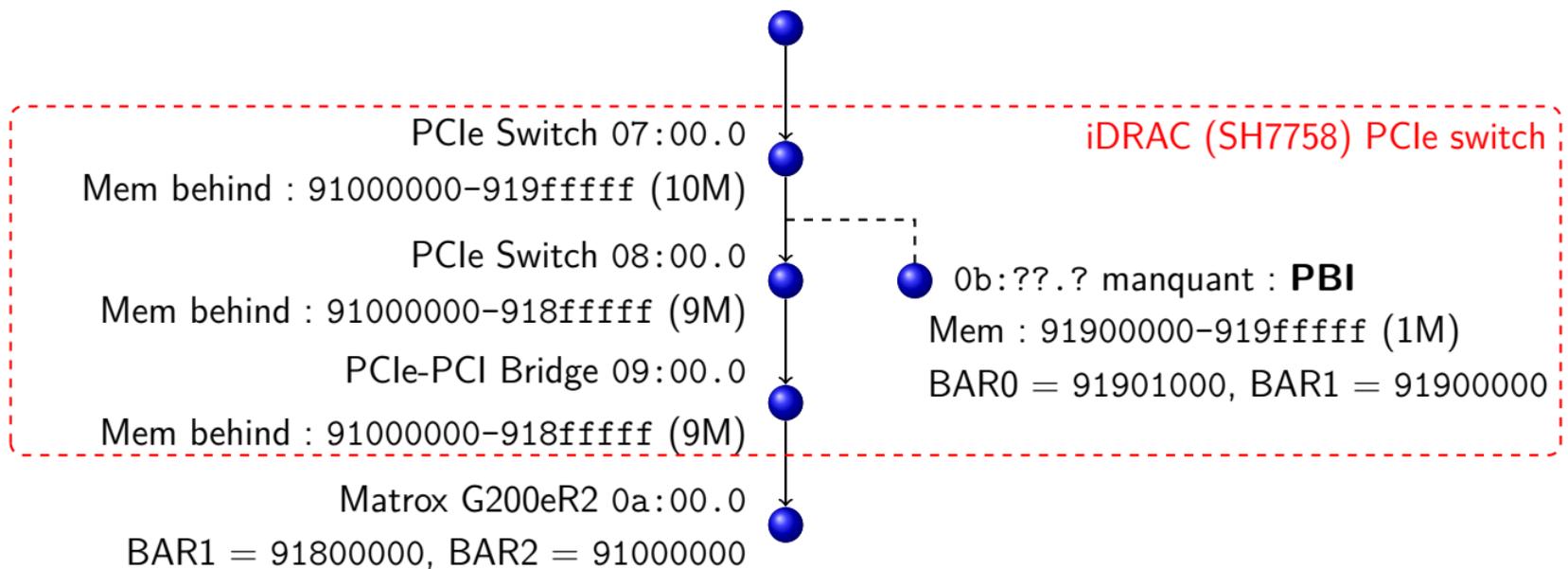
PCIe Root Port #8 00:1c.7 (Memory behind bridge : 91000000-919fffff (10M))





Arborescence PCIe avec les adresses

PCIe Root Port #8 00:1c.7 (Memory behind bridge : 91000000-919ffffff (10M))





À quoi ça sert ?

- ▶ Le PBI est un périphérique PCIe.
- ▶ Module noyau dans l'iDRAC : mémoire partagée et *mailbox* pour recevoir des commandes.
- ▶ `/usr/lib/libpbidrv.so.1.2.3` : couche d'abstraction pour les programmes de l'iDRAC.
- ▶ `/usr/include/pbidrv/lib_pbidrv.h` : structures du PBI (`MB_MSG_HDR`, `MB_MSG`, etc.).



À quoi ça sert ?

- ▶ Le PBI est un périphérique PCIe.
- ▶ Module noyau dans l'iDRAC : mémoire partagée et *mailbox* pour recevoir des commandes.
- ▶ `/usr/lib/libpbidrv.so.1.2.3` : couche d'abstraction pour les programmes de l'iDRAC.
- ▶ `/usr/include/pbidrv/lib_pbidrv.h` : structures du PBI (`MB_MSG_HDR`, `MB_MSG`, etc.).

```
// Mailbox protocol types
```

```
#define MSG_TYPE_IPMI    0x00
#define MSG_TYPE_CEM     0x08
#define MSG_TYPE_LCD     0x10
```

```
// Host to iDRAC LCD message IDs
```

```
#define LCD_MSG_INFO     0x00
#define LCD_MSG_COLOR    0x01
#define LCD_MSG_TEXT     0x02
#define LCD_MSG_BITMAP   0x03
```



Où est le PBI ?

```
externalsrc/u-boot-idrac8/u-boot_B0/board/renesas/sh77571cr/sh77571cr.c
static void init_pcie_bridge(void)
{
    /* ... */
    if (!(readw(PCIEBRG_CTRL_H8S) & 0x0001))
        return;
    // On 13G systems, fix issue with hiding PBI device.
    // Writing to PSPPBCTL DRS[1:0] = '01b'
    if(is_sh7758())
        writel(0x00000100, 0xffd60080);
}
```

Le PBI est caché :(



Bridge PCIe ?

```
include/asm-sh/cpu_sh7757.h et init_pcie_bridge :
#define PCIEBRG_BASE      0xffd60000
#define PCIEBRG_CTRL_H8S (PCIEBRG_BASE + 0x00)
#define PCIEBRG_CP_ADDR  (PCIEBRG_BASE + 0x10)
#define PCIEBRG_CP_DATA  (PCIEBRG_BASE + 0x14)
#define PCIEBRG_CP_CTRL  (PCIEBRG_BASE + 0x18)
writew(0xa501, PCIEBRG_CTRL_H8S); /* reset */
writew(0x0000, PCIEBRG_CP_CTRL);
writew(0x0000, PCIEBRG_CP_ADDR);
for (i = 0; i < pcie_cnt; i += 2) {
    tmp = (data[i] << 8) | data[i + 1];
    writew(tmp, PCIEBRG_CP_DATA);
}
writew(0xa500, PCIEBRG_CTRL_H8S); /* start */
printf("PCIe: Bridge loaded with 0x%x bytes\n", pcie_cnt);
```



Bridge PCIe

Le programme de démarrage de l'iDRAC (U-Boot) envoie le microcode/*firmware* du contrôleur d'un *bridge PCIe*.

```
externalsrc/u-boot-idrac8/u-boot_B0/board/renesas/sh77571cr/bridge7758.mot  
(format Motorola S-Record).
```



Bridge PCIe

Le programme de démarrage de l'iDRAC (U-Boot) envoie le microcode/*firmware* du contrôleur d'un *bridge PCIe*.

```
externalsrc/u-boot-idrac8/u-boot_B0/board/renesas/sh77571cr/bridge7758.mot  
(format Motorola S-Record).
```

```
objcopy -I srec -O binary bridge7758.mot bridge7758.bin
```



Plan

- 1 Introduction
- 2 Approche logicielle
- 3 Composants matériels spécifiques : CPLD, PBI, etc.
- 4 Le chaînon manquant



Contenu de bridge7758.mot

- ▶ Analyse d'entropie : pas de texte, mais pas chiffré/compressé
- ▶ On retrouve les *Vendor ID* / *Device ID* PCIe des switches et bridges!
- ▶ Quelle architecture ?
- ▶ `cpu_rec` : non reconnue.
- ▶ Appel à un ami.



Contenu de bridge7758.mot

- ▶ Analyse d'entropie : pas de texte, mais pas chiffré/compressé
- ▶ On retrouve les *Vendor ID* / *Device ID* PCIe des switches et bridges!
- ▶ Quelle architecture?
- ▶ `cpu_rec` : non reconnue.
- ▶ Appel à un ami.

C'est du H8S!

Ajout dans `cpu_rec`;) https://github.com/airbus-seclab/cpu_rec/issues/4



H8S en quelques lignes

- ▶ Famille H8 de Hitachi et Renesas (H8/300, H8/300H, H8/500...)
 - ▶ H8S dans certains *Embedded Controller Firmware* de batteries
 - ▶ H8/300H dans la brique LEGO^(R) Mindstorms RCX
 - ▶ apparemment aussi dans des climatisations
- ▶ *Lenovo System x3300 M4 Type 7382 server*
 - ▶ <https://systemx.lenovofiles.com/help/index.jsp?topic=%2Fcom.lenovo.sysx.7382.doc%2Fintroduction.html>
 - ▶ « on-board iBMC, Renesas SH7757 (IPMI 2.0) w/ RTMM H8S-2117A for system management. »
- ▶ Instructions : 2, 4, 6, 8 ou 10 octets
- ▶ Adressage sur 24 bits (16 Mo, incompatible avec le CPU *H8S* d'IDA)



H8S étudié

Organisation mémoire du microcontrôleur du bridge PCIe de l'iDRAC :

- ▶ 0x000000 à 0x017fff : microcode (96 Ko)
- ▶ 0xffa000 à 0xffbfff : RAM (8 Ko), pile décroissant à partir de la fin
- ▶ 0xffc000 à 0xffffffff : registres matériels (MMIO)



Communication iDRAC-H8S

Au début du microcode, des vecteurs d'interruptions.

```
if (*(u8*)0xffd033 & 1) { // Command trigger
    *(u8*)0xffd033 |= 1;
    u16 cmd = *(u16*)0xffd034; // Retrieve command
    if (cmd <= 0x7ae) *(u16*)0xffd022 = *(u16*)(0xffc000 + cmd);
    else if (cmd == 0x800) *(u16*)0xffd022 = *(u16*)0x000004; // => 0300
    else if (cmd == 0x802) *(u16*)0xffd022 = *(u16*)0x000006; // => 0301
    else if (cmd == 0x900) *(u16*)0xffd022 = *(u16*)0xffa9be;
    else if ((cmd & 0xff) <= 0x157) {
        u32 val = *(u32*)(0xffa000 + ((cmd>>12)-1)*0x180 + (cmd&0xffc));
        *(u16*)0xffd022 = (cmd & 2) ? (val >> 16) : (val & 0xffff);
    }
}
```



Communication iDRAC-H8S

```
# H8S ffd034 = iDRAC ffd60034
```

```
$ MemAccess2 -ww -c 1 -a 0xffd60034 -d 0802  
MemAccess OK !
```

```
# H8S ffd033 = iDRAC ffd60030
```

```
$ MemAccess2 -ww -c 1 -a 0xffd60030 -d 0001  
MemAccess OK !
```

```
# H8S ffd022 = iDRAC ffd60028
```

```
$ MemAccess2 -rw -c 1 -a 0xffd60028
```

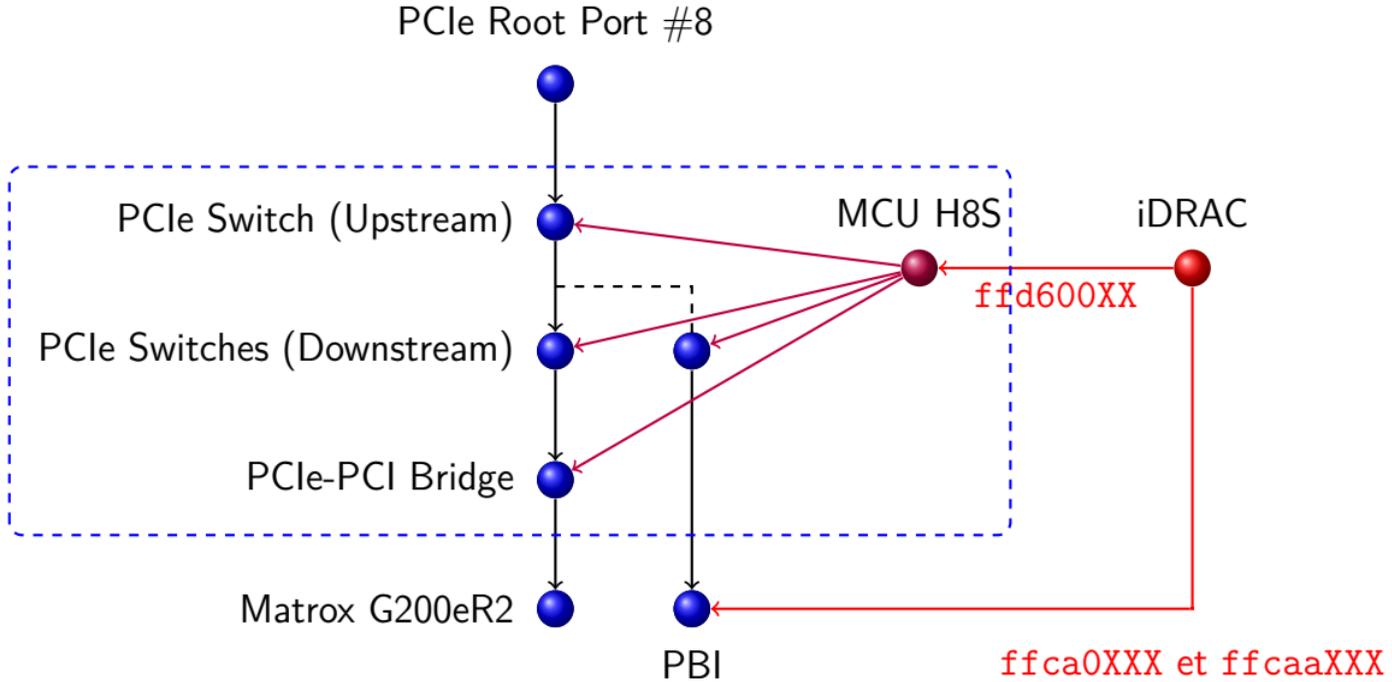
```
+          0    2    4    6 :    8    A    C    E  
-----
```

```
0xffd60028 = 0301
```

```
MemAccess OK !
```



Connexions





Conclusion

Un attaquant qui gagne l'accès à un iDRAC peut effectuer des actions similaires à l'accès physique au serveur.

Est-ce qu'un attaquant peut accéder au contenu de la mémoire principale ?

- ▶ Il peut connecter un périphérique USB virtuel (clavier, interface réseau, etc.).
- ▶ Il peut reprogrammer le microcontrôleur qui pilote le bus PCIe entre la carte graphique et le système principal.
- ▶ Il est probable qu'il puisse réactiver un périphérique PCIe *PBI*.
- ▶ Il est probable qu'il puisse émettre des requêtes DMA depuis ce microcontrôleur.



Conclusion

Un attaquant qui gagne l'accès à un iDRAC peut effectuer des actions similaires à l'accès physique au serveur.

Est-ce qu'un attaquant peut accéder au contenu de la mémoire principale ?

- ▶ Il peut connecter un périphérique USB virtuel (clavier, interface réseau, etc.).
- ▶ Il peut reprogrammer le microcontrôleur qui pilote le bus PCIe entre la carte graphique et le système principal.
- ▶ Il est probable qu'il puisse réactiver un périphérique PCIe *PBI*.
- ▶ Il est probable qu'il puisse émettre des requêtes DMA depuis ce microcontrôleur.

Quoiqu'il en soit, il est recommandé de ne pas exposer un iDRAC sur Internet.

Plus de recommandations :

<https://www.cert.ssi.gouv.fr/actualite/CERTFR-2017-ACT-014/>



Questions

?



Le chaînon manquant

Show me the code !

Bientôt le CPU H8S pour Ghidra sur <https://github.com/idrackar>



Commandes IPMI pour le pilotage de l'écran LCD (1/2)

- ▶ How can I set a custom text on the LCD display on Dell PowerEdge servers?
[https://serverfault.com/questions/81015/
how-can-i-set-a-custom-text-on-the-lcd-display-on-dell-poweredge-servers](https://serverfault.com/questions/81015/how-can-i-set-a-custom-text-on-the-lcd-display-on-dell-poweredge-servers)
- ▶ [Ipmitool-devel] Fwd : LCD access on PowerEdge 1950 <https://www.mail-archive.com/ipmitool-devel@lists.sourceforge.net/msg00352.html>
- ▶ opensource.dell.com :
`ipk-dropbox/persmod/image/etc/sysapps_script/pm_lcd_update.sh`



Commandes IPMI pour le pilotage de l'écran LCD (2/2)

```
% ipmitool -U user -P pass -L ADMINISTRATOR raw \  
0x6 0x58 0xc1 0 0 11 0x48 0x69 0x2c 0x53 0x53 0x54 0x49 0x43 0x20 0x3b 0x29  
% ipmitool -U user -P pass -L ADMINISTRATOR raw \  
0x6 0x58 0xc2 0
```

Documentation :

- ▶ 0x6 : NetFn = Applications
- ▶ 0x58 : Operation = set system information
- ▶ 0xc1/0xc2 : set the string to this / show this string to LCD
- ▶ 0 : which chunk of 16 bytes is edited
- ▶ 0 : encoding = ANSSI
- ▶ 11 : length
- ▶ text (max. 14 characters according to several web pages)



iDRAC 8 vs. iDRAC 9

	iDRAC 8	iDRAC 9
Linux, U-Boot, etc.	Oui	Oui
Sur opensource.dell.com ?	Oui	Oui
CPU	SuperH4	ARM
SELinux ?	Non	Oui
PBI ?	Oui	Non
Compte initial	root/calvin	Indiqué sur une étiquette



Les CPU de Renesas

MCU for Every Application

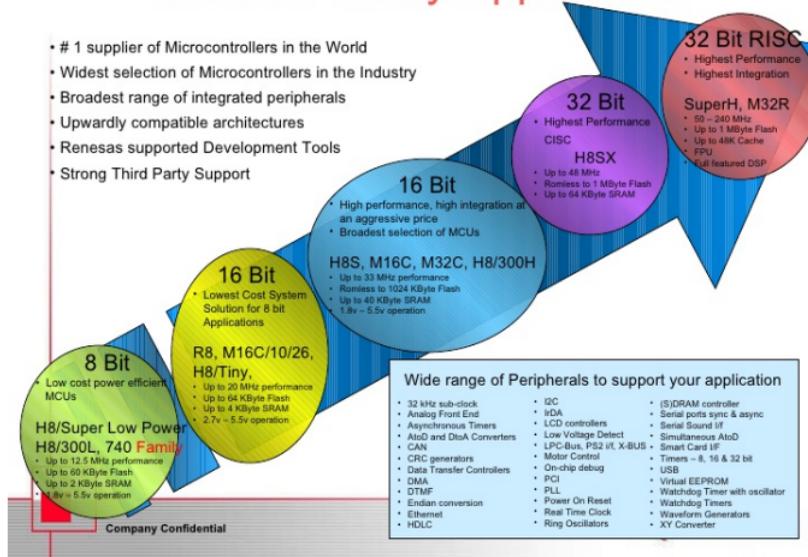


FIGURE 16 – Source : <https://www.slideshare.net/Flashdomain/4-r8c-v3ppt>