



---

(Inventar) Daten sammeln



# About me

---

- Roger Zander
  - ROMAWO.com / itnetX
  - Ex MVP (2012-2022)
  - Workplace-Enthusiast
  - Bekennder Minimalist und Pragmatiker (es lebe die Automatisierung !!)
  - ...



# Wozu

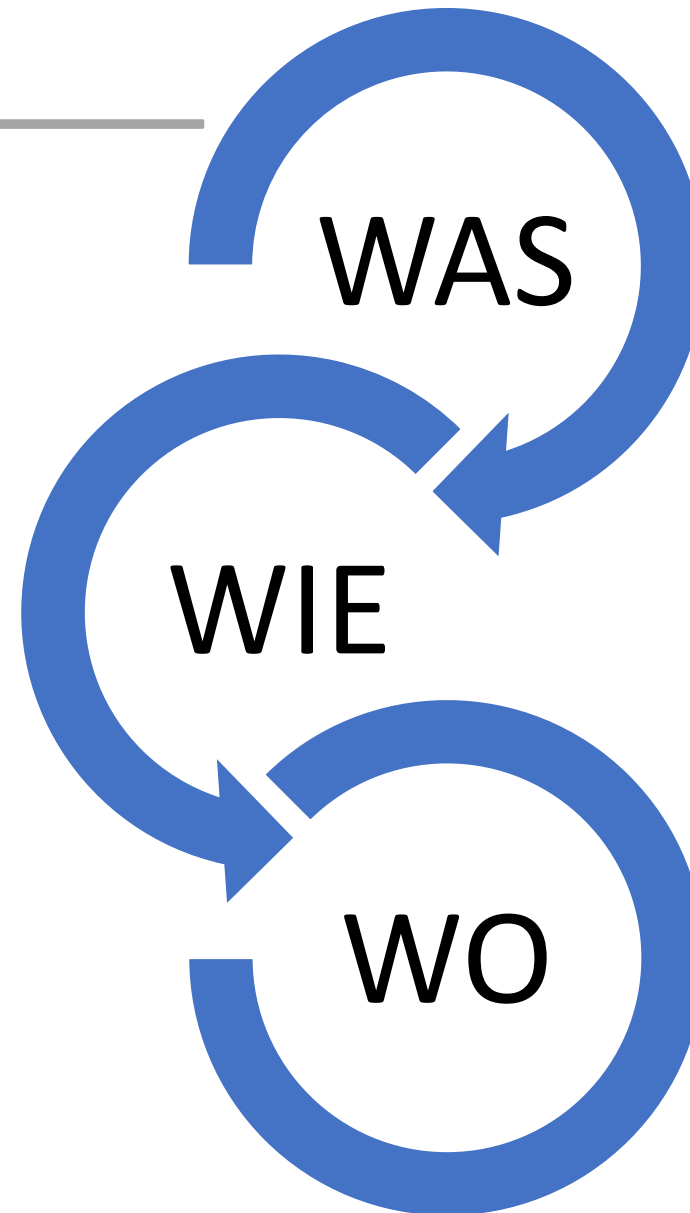
---

- Die Inventar-Möglichkeiten von Intune sind “bescheiden”
- Vertrauen ist gut, Kontrolle ist besser... (Konfigurationen / Settings)
- Erhebung zusätzlicher Daten
- ...



# Übersicht

Bevor man eine Lösung baut, nicht nur das “**Was**” klären, sondern **vorgängig** auch das “**Wie**” und “**Wo**”... !!!



 Scripten

 Scripten



# Was (1)

---

- Was ist der Grund, weshalb die Daten benötigt werden... ?
- Welche Informationen werden benötigt? Beispiele:
  - Software Inventar
  - Bildschirm SerieNr.
  - Garantie-Daten der Geräte
  - Lokale Administratoren
  - Defender / Firewall Einstellungen
  - Windows/Defender Events
  - ...
- Sind die Informationen nicht schon irgendwo vorhanden?
- Werden personen-bezogene Daten ermittelt ? -> Datenschutz



## Was (2)

---

- Was genau...
  - Welche Informationen/Attribute werden benötigt
  - Was ist das Key-Attribut
    - Computername
    - Serie Nr.
    - UUID
    - Username
    - UPN
    - ...



# Was Beispiel: Bildschirm-Serie-Nr

---

- Attribute:
  - Serie-Nr , Modell, Diagonale, Alter, Computer
- Key:
  - Key = Computername?
    - Was ist, wenn der Computer und der Bildschirm nicht fix verbunden sind (Docking-Station)
  - Key = Bildschirm SerieNr.?
    - Was wenn ein Bildschirm keine SerieNr hat (e.g. Laptop Displays)



# Was Beispiel: Inventar

---

- Computer Inventar:
  - Key = Computername ?
    - Was ist mit doppelten Computernamen
    - Wie verhält sich der Computername wenn das Gerät neu installiert wird.
    - Namenskonzepte ändern sich... und dann ?
- Computer & Benutzer Inventar:
  - z.B. Software aus dem Benutzerkontext
  - Key = Computername + Benutzername ? Oder doch nur Computername?
    - Was wenn der Benutzer wechselt oder nicht mehr aktiv ist ?
    - Benutzer als zusätzliches Attribut ?





# Wie (1)

---

- Wie können die Daten elektronisch ermittelt werden (Script/API)
- In welchem Format werden die Daten benötigt? (JSON/XML/CSV/..)
  - Wie werden die Daten weiterverarbeitet ? (PowerBI, Import in ..., ... )
- Wie oft müssen die Daten erhoben werden?
- Wie viele Daten werden generiert (Datenmenge)
- Wie lange müssen/dürfen die Daten gespeichert werden
- Zeitpunkt der Datenerhebung (egal, beim aufstarten, 8:00AM, ...)

Hinweis: Den generierten Workload nicht unterschätzen... Wenn einige 100/1000 Geräte fast gleichzeitig starten, kann es erhebliche Last generieren...



# Wie: Inventar mit PowerShell

```
Get-CimInstance win32 ComputerSystemProduct -Namespace root\cimv2 |  
Select-Object -ExcludeProperty cim*, pscomp* | ConvertTo-Json  
{  
  "Name": "20F0BT01QW",  
  "Version": "ThinkStation P620",  
  "Caption": "Computer System Product",  
  "Description": "Computer System Product",  
  "IdentifyingNumber": "EN00ARQD",  
  "SKUNumber": null,  
  "Vendor": "LENOVO",  
  "UUID": "28FAF4DE-602A-ED21-8100-047BCBA56815"  
}
```



# Wie: Inventar mit PowerShell (2)

---

- PowerShell Versionen beachten (PS5.1 vs PS7.2/PS7.3)
  - PS5.x = .NET 4.8
    - Get-WmiObject
    - Get-CimInstance
  - PS7.2 = .NET6 (LTSC)
    - Get-CimInstance
  - PS7.3 = .NET7
    - Get-CimInstance
  - *PS7.4 = .NET8 (LTSC) ?*

<https://learn.microsoft.com/en-us/powershell/scripting/whats-new/differences-from-windows-powershell?view=powershell-7.3>



## Wie: Inventar mit PowerShell (3)

---

- <https://github.com/rzander/DocumentDB-Inventory>
  - Erstellt Inventar und speichert es in CosmosDB
- <https://github.com/rzander/jaindb/blob/master/examples/Windows%20Inventory/inventory.ps1>
  - JainDB bietet verschiedene Storage Provider an (File, Blob, CosmosDB etc. )



# Wie: Bildschirm SN mit PowerShell

```
Get-CimInstance wmiMonitorID -Namespace root\wmi
Active           : True
InstanceName     : 
DISPLAY\PHL0939\5&15a9bd37&0&UID397569_0
ManufacturerName : {80, 72, 76, 0...}
ProductCodeID    : {48, 57, 51, 57...}
SerialNumberID   : {49, 56, 49, 48...}
UserFriendlyName : {80, 72, 76, 32...}
UserFriendlyNameLength : 13
WeekOfManufacture : 42
YearOfManufacture : 2019
PSComputerName   :
```



## Wie: Bildschirm SN mit PowerShell (2)

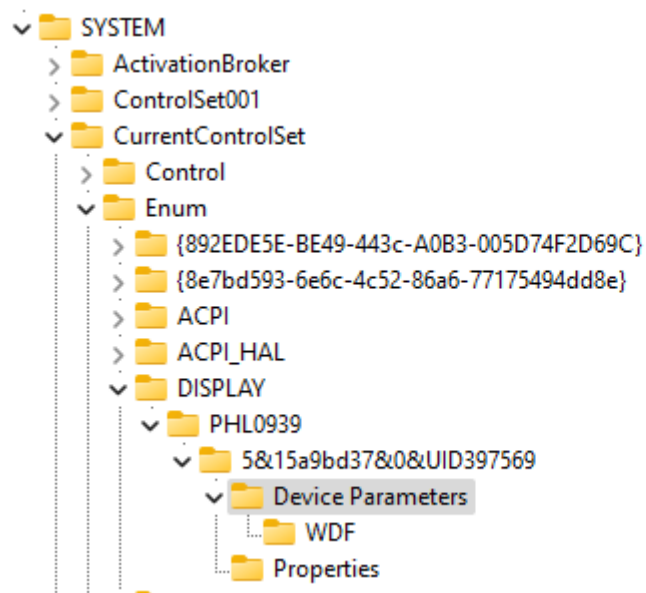
---

- **Get Manufacturer:**  
`[System.Text.Encoding]::UTF8.GetString((Get-CimInstance wmiMonitorID -Namespace root\wmi).ManufacturerName)`
- **Extract SerialNumber:**  
`[System.Text.Encoding]::UTF8.GetString((Get-CimInstance wmiMonitorID -Namespace root\wmi).SerialNumberID)`
- **UserFriendlyName:**  
`[System.Text.Encoding]::UTF8.GetString((Get-CimInstance wmiMonitorID -Namespace root\wmi).UserFriendlyName)`



# Wie: Bildschirm SN mit PowerShell (3)

- <https://github.com/rzander/MonitorDetails>
- Decode EDID (Extended Display Identification Data)
  - [https://en.wikipedia.org/wiki/Extended\\_Display\\_Identification\\_Data](https://en.wikipedia.org/wiki/Extended_Display_Identification_Data)



Name	Type	Data
(Default)	REG_SZ	(value not set)
EDID	REG_BINARY	00 ff ff ff ff ff 00 41 0c 39 09 12 07 00 00 2a 1d 01 03 80 69 21 78 3f 7f 25 ad 4f a5 2



# Wie: Bildschirm SN mit PowerShell (4)

- <https://people.freedesktop.org/~imirkin/edid-decode/>

EDID  No file chosen

```
00 ff ff ff ff ff 00 41 0c 39 09 12 07 00 00 2a 1d 01 03 80 69 21 78 3f 7f
25 ad 4f 47 a5 26 10 50 54 bf ef 00 d1 c0 b3 00 95 00 81 80 81 40 81 c0 d1 00
01 01 68 73 00 a0 f0 b0 1f 40 30 40 35 00 1c 49 41 00 00 1a 58 c0 00 a0 f0 b0
1f 40 30 40 3a 00 1c 49 41 00 00 1a 00 00 00 fc 00 50 48 4c 20 34 33 39 50 39
48 0a 20 20 00 00 00 fd 00 30 64 1e 96 3c 00 0a 20 20 20 20 20 20 01 88 02 03
42 f1 4d 61 60 1f 90 14 01 03 04 12 13 05 02 40 23 09 07 07 83 01 00 00 67 03
0c 00 10 00 38 40 67 d8 5d c4 01 78 80 03 68 1a 00 00 01 01 30 64 00 e3 05 e3
01 e3 0f 03 00 e6 06 07 01 61 56 1c a2 4e 00 a0 a0 b0 23 40 30 20 58 04 1c 49
41 00 00 1a a4 29 00 a0 50 b0 23 40 30 20 58 04 1c 49 41 00 00 1a 28 3c 80 a0
70 b0 23 40 30 20 36 00 1c 49 41 00 00 1a 00 00 00 00 00 00 00 93
```

[edid-decode.git](https://people.freedesktop.org/~imirkin/edid-decode/)

Parsed

```
Block 0, Base EDID:
EDID Structure Version & Revision: 1.3
Vendor & Product Identification:
Manufacturer: PHL
Model: 2361
Serial Number: 1810
Made in: week 42 of 2019
Basic Display Parameters & Features:
Digital display
Maximum image size: 105 cm x 33 cm
Gamma: 2.20
DPMS levels: off
Undefined display color type
Default (sRGB) color space is primary color space
First detailed timing is the preferred timing
Supports GTF timings within operating range
Color Characteristics:
Red : 0.6767, 0.3115
Green: 0.2802, 0.6474
Blue : 0.1484, 0.0644
White: 0.3134, 0.3291
Established Timings I & II:
IBM : 720x400 70.081663 Hz 9:5 31.467 kHz 28.320000 MHz
DMT 0x04: 640x480 59.940476 Hz 4:3 31.469 kHz 25.175000 MHz
Apple : 640x480 66.666667 Hz 4:3 35.000 kHz 30.240000 MHz
DMT 0x05: 640x480 72.808802 Hz 4:3 37.861 kHz 31.500000 MHz
DMT 0x06: 640x480 75.000000 Hz 4:3 37.500 kHz 31.500000 MHz
DMT 0x08: 800x600 56.250000 Hz 4:3 35.156 kHz 36.000000 MHz
DMT 0x09: 800x600 60.316541 Hz 4:3 37.879 kHz 40.000000 MHz
DMT 0x0a: 800x600 72.187572 Hz 4:3 48.077 kHz 50.000000 MHz
DMT 0x0b: 800x600 75.000000 Hz 4:3 46.875 kHz 49.500000 MHz
Apple : 832x624 74.551266 Hz 4:3 49.726 kHz 57.284000 MHz
DMT 0x10: 1024x768 60.003840 Hz 4:3 48.363 kHz 65.000000 MHz
DMT 0x11: 1024x768 70.069359 Hz 4:3 56.476 kHz 75.000000 MHz
DMT 0x12: 1024x768 75.028582 Hz 4:3 60.023 kHz 78.750000 MHz
DMT 0x24: 1280x1024 75.024675 Hz 5:4 79.976 kHz 135.000000 MHz
Standard Timings:
DMT 0x52: 1920x1080 60.000000 Hz 16:9 67.500 kHz 148.500000 MHz
DMT 0x3a: 1680x1050 59.954250 Hz 16:10 65.290 kHz 146.250000 MHz
DMT 0x2f: 1440x900 59.887445 Hz 16:10 55.935 kHz 106.500000 MHz
DMT 0x23: 1280x1024 60.019740 Hz 5:4 63.981 kHz 108.000000 MHz
DMT 0x20: 1280x960 60.000000 Hz 4:3 60.000 kHz 108.000000 MHz
DMT 0x55: 1280x720 60.000000 Hz 16:9 45.000 kHz 74.250000 MHz
DMT 0x45: 1920x1200 59.884600 Hz 16:10 74.556 kHz 193.250000 MHz
Detailed Timing Descriptors:
DTD 1: 3840x1200 60.000000 Hz 16:5 73.860 kHz 295.440000 MHz (1052 mm x 329 mm)
Hfront 48 Hsync 64 Hback 48 Hpol P
Vfront 3 Vsync 5 Vback 23 Vpol N
DTD 2: 3840x1200 100.000000 Hz 16:5 123.100 kHz 492.400000 MHz (1052 mm x 329 mm)
Hfront 48 Hsync 64 Hback 48 Hpol P
Vfront 3 Vsync 10 Vback 18 Vpol N
Display Product Name: 'PHL 439P9H'
Display Range Limits:
Monitor ranges (GTF): 48-100 Hz V, 30-150 kHz H, max dotclock 600 MHz
Extension blocks: 1
Checksum: 0x88
```





## Wie: Bildschirm SN mit PowerShell (5)

---

Aber:

- Nicht alle Hersteller implementieren den Standard gleich
- Behauptung: Monitor Treiber muss installiert sein damit EDID Daten ausgelesen werden.
- Bildschirme sind nicht zwingend an Computer gebunden (Computer ist nicht der Key).
- Je nach Hersteller muss für den Key die SN ergänzt werden (Hersteller + ProductCode + Jahr + Week + SN = PHL-0939-2019-42-1810)



# Wie: Bildschirm SN mit PowerShell (6)

PartitionKey	RowKey	Timestamp	M...	Name	MonitorId	HSize	VSize	Serial	PnPDeviceID	Week...	Year...	Computername
...	09D12015321573	2023-08-10T16:15:04.1675994Z	09D1	BenQ GL2460	BNQ78CE	53	30	21573	DISPLAY\BNQ78CE\4&2E923345...	3	2015	...
...	10AC20163313378	2023-08-17T14:20:08.3902606Z	10AC	Dell P2317H (DP)	DEL40F3	51	29	13378	DISPLAY\DEL40F3\5&1252189&0...	33	2016	...
...	10AC20163317986	2023-08-17T14:20:08.5781531Z	10AC	Dell P2317H (DP)	DEL40F3	51	29	17986	DISPLAY\DEL40F3\5&22B844B7&...	33	2016	...
...	10AC20163319778	2023-08-10T07:12:02.4193653Z	10AC	Dell P2317H (DP)	DEL40F3	51	29	19778	DISPLAY\DEL40F3\4&1255F2D5...	33	2016	...
...	10AC20182116983	2023-08-17T14:20:21.0020688Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	16983	DISPLAY\DELD0D6\5&1252189&...	21	2018	...
...	10AC20182122615	2023-08-10T11:55:08.0606755Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	22615	DISPLAY\DELD0D6\4&94106C7&...	21	2018	...
...	10AC20185014146	2023-08-17T14:20:08.5351785Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	14146	DISPLAY\DELD0D6\5&322FD008...	50	2018	...
...	10AC20193514156	2023-08-17T14:20:08.395258Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	14156	DISPLAY\DELD0D6\5&229FD66E...	35	2019	...
...	10AC20193514668	2023-08-17T14:20:20.9051239Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	14668	DISPLAY\DELD0D6\5&229FD66E...	35	2019	...
...	10AC20204619266	2023-08-17T14:20:08.3932594Z	10AC	DELL P2319H(DisplayPort)	DELD0D6	51	29	19266	DISPLAY\DELD0D6\5&21202824...	46	2020	...
...	10AC20131317740	2023-08-17T14:20:08.39126Z	10AC	Dell U2713HM (DP)	DEL4080	60	34	17740	DISPLAY\DEL4080\5&2CD8665B...	13	2013	...
...	10AC20131320556	2023-08-17T14:20:08.3932594Z	10AC	Dell U2713HM (DP)	DEL4080	60	34	20556	DISPLAY\DEL4080\5&322FD008...	13	2013	...
...	09D120213257	2023-08-09T17:10:32.9910371Z	09D1	Generic Monitor (BenQ GW2780)	BNQ78E6	60	34	257	DISPLAY\BNQ78E6\4&1BD78084...	3	2021	...
...	10AC20194014156	2023-08-17T14:20:08.5651612Z	10AC	Generic Monitor (DELL P2720DC)	DELD0FC	60	34	14156	DISPLAY\DELD0FC\5&229FD66E...	40	2019	...
...	10AC20203914675	2023-08-17T14:20:21.1010129Z	10AC	Generic Monitor (DELL P2720DC)	DELD0FC	60	34	14675	DISPLAY\DELD0FC\5&229FD66E...	39	2020	...
...	10AC20204318259	2023-08-17T14:20:08.5571656Z	10AC	Generic Monitor (DELL P2720DC)	DELD0FC	60	34	18259	DISPLAY\DELD0FC\5&21202824...	43	2020	...
...	10AC20211612610	2023-08-17T14:20:08.3922596Z	10AC	Generic Monitor (DELL P2720DC)	DELD0FC	60	34	12610	DISPLAY\DELD0FC\4&1255F2D5...	16	2021	...
...	10AC20211620802	2023-08-17T14:20:08.3892614Z	10AC	Generic Monitor (DELL P2720DC)	DELD0FC	60	34	20802	DISPLAY\DELD0FC\4&1255F2D5...	16	2021	...



# Wo (1)

---

Script läuft als  
SYSTEM !!!

- Azure FileShare (\$0.03/GB)
  - Passwort/Key muss im Script hinterlegt werden
    - Vollzugriff auf alle Daten im FileShare
  - Verbindungsaufbau ist langsam
  - Es können grosse Datenmengen gespeichert werden.
  - Performance (standard): 20'000 IOPS; 10 write operations per second; 60MiB/sec
  - Auswertung kompliziert, da Datei-basierend

<https://learn.microsoft.com/en-us/azure/storage/files/storage-files-scale-targets>



## Wo (1)

---

```
net use [drive letter]  
\\server.file.core.windows.net\sharename  
/u:account  
Lz4A3tX1WDFZx0XKH1DiFrN1TFjOddPls9QgmpRLQPladAxa  
XnVPcaDJQAWzdyiQJT3BhmQ+4Oo1S7Fg4axxrg==
```



## Wo (2)

---

- Azure Blob Storage (\$0.02/GB)
  - SAS Token muss im Script hinterlegt werden
    - Token kann nur schreibrechte haben, ohne dass damit Daten ausgelesen werden können.
    - Token kann zeitlich limitiert werden.
  - Verbindungsaufbau ist schnell (HTTP Post)
  - Gute performance: “up to 500 requests per second; up to 60 MiB per second”
  - Auswertung kompliziert, da Datei-basierend (PowerBI unterstützt Azure Blob Storage)

<https://learn.microsoft.com/en-us/azure/storage/blobs/scalability-targets>

<https://rzander.azurewebsites.net/upload-file-to-azure-blob-storage-with-powershell/>



## Wo (2)

---

### URL mit SAS Token:

```
https://server.table.core.windows.net/tablename?  
sv=2019-02-02&st=2023-12-  
11T14%3A21%3A00Z&se=2023-12-  
12T14%3A21%3A00Z&sp=r&sig=%2F2jH0v2s184zdeno05i7  
CL64GWdN1AI9r7kX1DDWA9w%3D&tn=tablename
```



## Wo (3)

---

- Azure Table Storage (\$0.08/GB)
  - SAS Token muss im Script hinterlegt werden
    - Token kann nur schreibrechte haben, ohne dass damit Daten ausgelesen werden können.
    - Token kann zeitlich limitiert werden.
  - Verbindungsaufbau ist schnell (HTTP POST)
  - Gute performance: “up to 20’000 transactions per second; throughput up to 2’000 entities per second”
  - Entity size Limit 1MiB, 255 properties (incl. Timestamp and Partition- / Row-Key)
  - Incremental Updates sind möglich (HTTP PUT/PATCH)
  - PowerBi kann direkt auf die Daten zugreifen (mit SAS Token werden standardmässig nur die ersten 1000 Objekte zurückgegeben)

<https://learn.microsoft.com/en-us/azure/storage/tables/scalability-targets>

<https://rzander.azurewebsites.net/bitlocker-management-with-azure-storage-table/>



# Wo Beispiel: Azure Table Storage

## Lenovo Warranty Data:

PartitionKey	RowKey	Timestamp	Country	Description	ID	InWarranty	Name	Product	Purchased
lenovo	EM08...	2023-10-26T15:58:29.8488397Z	CH	This product has a three year limi...	36Y,UKN	true	3Y On-site, 9X5,3Y Premier Supp...	WORKSTATIONS/THINKSTATION-...	2023-03-15
lenovo	EM08...	2023-10-31T05:15:18.9086219Z	CH	This machine is entitled to a warr...	UKN,36Y	true	3Y Premier Support,3Y On-site, 9...	WORKSTATIONS/THINKSTATION-...	2023-03-30
lenovo	PC0...	2023-04-25T11:52:15.2389397Z	CH	This product has a three year limi...	36Y	false	3Y On-site, 9X5	DESKTOPS-AND-ALL-IN-ONES/T...	0001-01-01
lenovo	PC0...	2023-10-25T17:32:23.0592328Z	CH	The battery included within this p...	1YOBAT,36Y,AH1,TP2,TVU,UKL	true	1YR Battery (On-site Warranty),3Y...	LAPTOPS-AND-NETBOOKS/THIN...	2018-06-15
lenovo	PC0...	2023-10-25T06:05:08.0738113Z	CH	This product has a four year limit...	TP2,36Y,1YOBAT	false	4Y On-site, 9X5,3Y On-site, 9X5,1...	LAPTOPS-AND-NETBOOKS/THIN...	0001-01-01
lenovo	PC0...	2023-10-24T13:14:27.2007326Z	CH	This product has a three year limi...	36Y,1YOBAT,TP2	false	3Y On-site, 9X5,1YR Battery (On-s...	LAPTOPS-AND-NETBOOKS/THIN...	0001-01-01
lenovo	PC0...	2023-10-25T04:47:28.8262092Z	CH	This product has a four year limit...	TP2,36Y,1YOBAT	false	4Y On-site, 9X5,3Y On-site, 9X5,1...	LAPTOPS-AND-NETBOOKS/THIN...	0001-01-01
lenovo	PC0...	2023-10-25T07:54:41.3733155Z	CH	This product has a four year limit...	TP2,1YOBAT,36Y	false	4Y On-site, 9X5,1YR Battery (On-s...	LAPTOPS-AND-NETBOOKS/THIN...	0001-01-01
lenovo	PC0...	2023-10-17T13:11:28.9168261Z	CH	The battery included within this p...	1YOBAT,36Y,AH1,TP2,TVU,UKL	true	1YR Battery (On-site Warranty),3Y...	LAPTOPS-AND-NETBOOKS/THIN...	2018-06-15

Microsoft Storage Explorer: <https://azure.microsoft.com/en-us/products/storage/storage-explorer/>





## Wo (4)

---

- Azure Log Analytics (\$3.20/GB)
  - **Immutable** (unveränderbar)
  - Es gibt keinen RowKey, die aktuellen Daten muss über ein Kusto-Query ermittelt werden.
  - Performance Limits „nur“ bei der Abfrage (Queries)
    - 200 queries per 30 seconds per user
  - Data Retentions default = 31 days
  - Abfragen der Daten generiert (geringe) Kosten.

<https://learn.microsoft.com/en-us/azure/azure-monitor/service-limits>

<https://learn.microsoft.com/en-us/azure/azure-monitor/logs/cost-logs>



## Wo (5)

---

- Azure Storage Queue (\$0.08/GB)
  - SAS Token muss im Script hinterlegt werden
    - Token kann nur schreibrechte haben, ohne dass damit Daten ausgelesen werden können.
    - Token kann zeitlich limitiert werden.
  - Verbindungsaufbau ist schnell (HTTP POST)
  - Gute performance: Throughput: “up to 2’000 messages per second”
  - Message Size limit: 64KiB
  - Auswertung der Daten nicht direkt möglich. Aber Services wie AzureFunction können Daten direct aus Queue beziehen.
  - TTL auf Message

<https://learn.microsoft.com/en-us/azure/storage/queues/scalability-targets>



## Wo (6)

---

- Azure CosmosDB (\$0.3/GB, + RU)
  - Schnelle, verteilte NoSQL- und relationale Datenbank in beliebigem Umfang
  - Access-Key muss im Script hinterlegt werden
    - Nur Read-Only oder Write Key für die ganze Instanz
  - Hoch skalierbar, Performance ist abhängig von RU/s limit.
  - TTL auf Document, Documentsize = 2MB (16MB)
  - Query basierte Abfragen z.B. direct via PowerBi

<https://learn.microsoft.com/en-us/azure/cosmos-db/concepts-limits>



## Wo (7)

---

- Azure Cache for Redis (C1=\$44 für 1GB, 1'000 connections)
  - Key muss im Script hinterlegt werden
  - Hoch performant
  - TTL auf Document

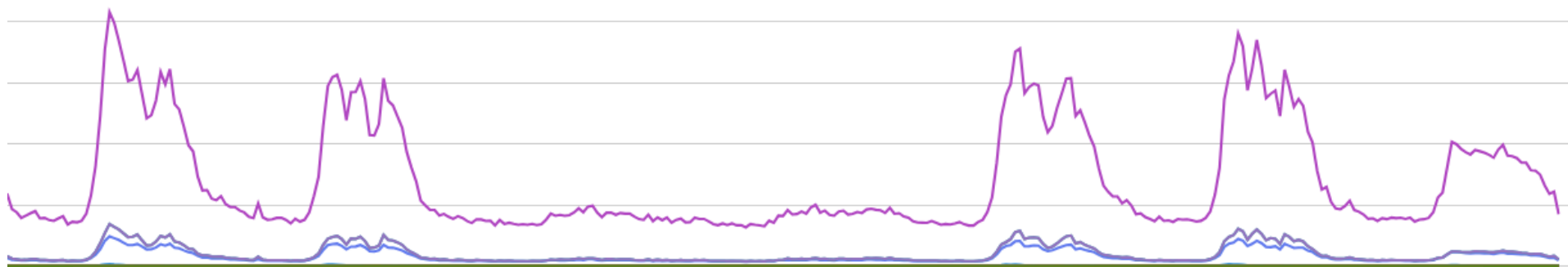
<https://azure.microsoft.com/en-us/products/cache/>

<https://azure.microsoft.com/en-us/pricing/details/cache/>



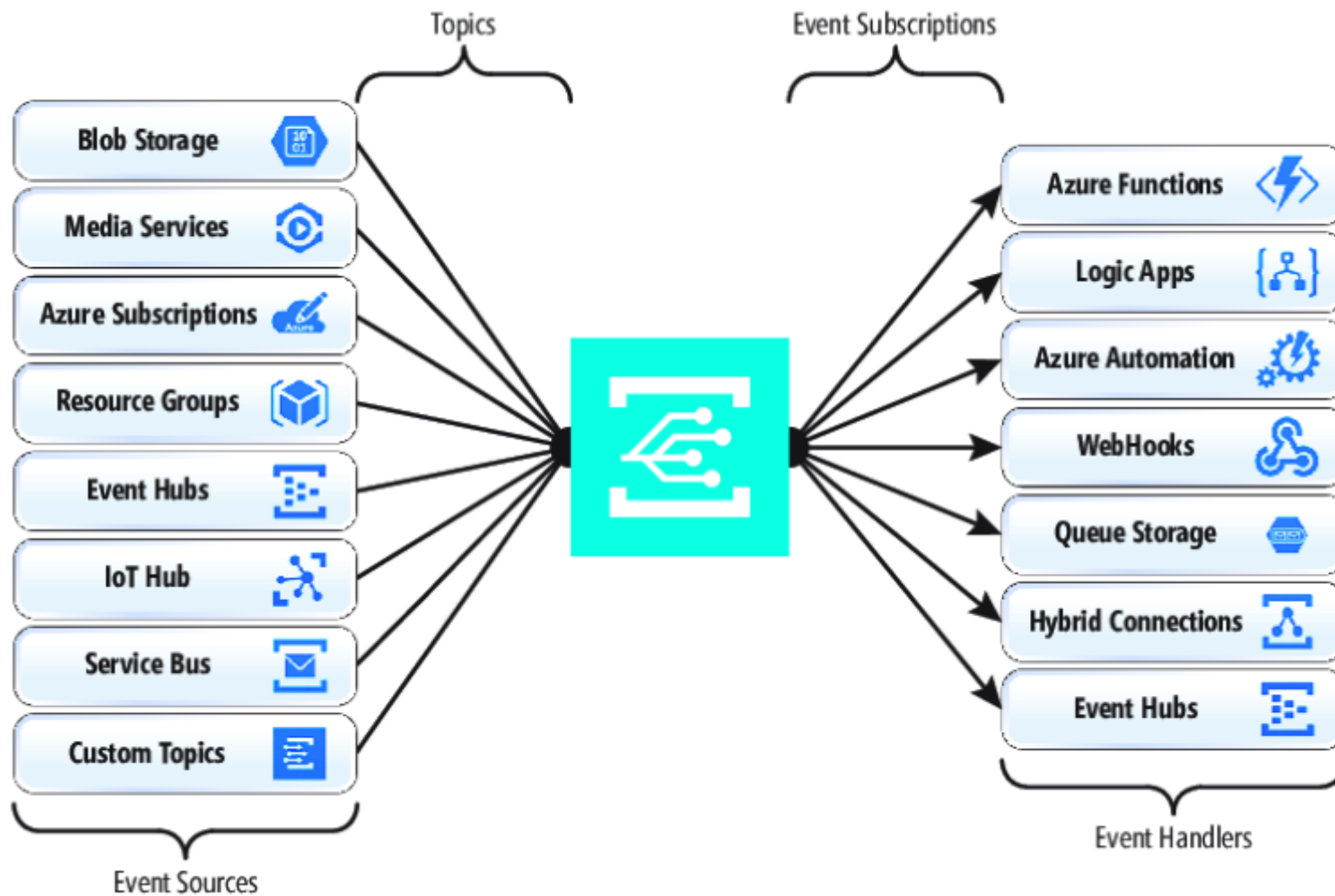
# Wo (8) Skalierung

- Storage Queue, CosmosDB und Redis eignen sich nur bedingt für eine direkte verwendung als Speicher. In diesen Fällen kann:
  - **Azure Event Grid** (\$0.6/1M operations)
  - **Azure Service Bus** (\$0.0135/h + \$0.8/1M operations ; 13M ops/month included)..die Daten entgegennehmen, via Azure Functions verarbeiten und schlussendlich in CosmosDB speichern/archivieren. Redis kann z.B. als Cache die Zugriffe auf CosmosDB reduzieren um RU(\$) bei Abfragen zu sparen.





# Wo (8): Azure Event Grid





# Übersicht

---

