PSCONF.EU

PowerShell Conference Europe

Basic Toolmaking: storing persistent data in PowerShell scripts

Evgenij Smirnov

VIENNA

20 - 23 June 2022

PSCONF.EU

PowerShell Conference Europe

Basic Toolmaking: storing persistent data in PowerShell scripts

Evgenij Smirnov

VIENNA

20 - 23 June 2022

Many thanks to our sponsors:

SquaredUp

au2mator

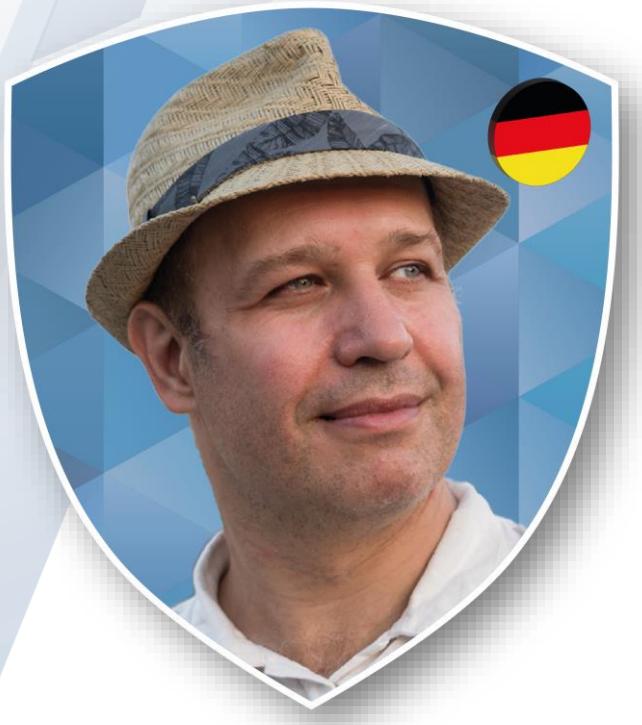
PATCH MY PC

Chocolatey

System Frontier

ScriptRunner®

@cj_berlin



```
PS> Get-SpeakerBio -Brief
FullName : Evgenij Smirnov
YearsInIT: 30
Location : Berlin
Employer : SVA GmbH
JobTitle : {Consultant,Architect}
UserGroup: {PSUGB,WSUG-B,EXUSG,VMUGB}
SpeakerAt: {PSConfEU,PSDAY.UK,CIM,…}
Twitter  : @cj_berlin
BlogURL  : https://it-pro-berlin.de
```

Microsoft® Most Valuable Professional

@cj_berlin

In this session

- Why do we need to store data in our scripts?
- Choosing the right data storage strategy
- Implementing data storage in PowerShell

Use Cases

Start with Why

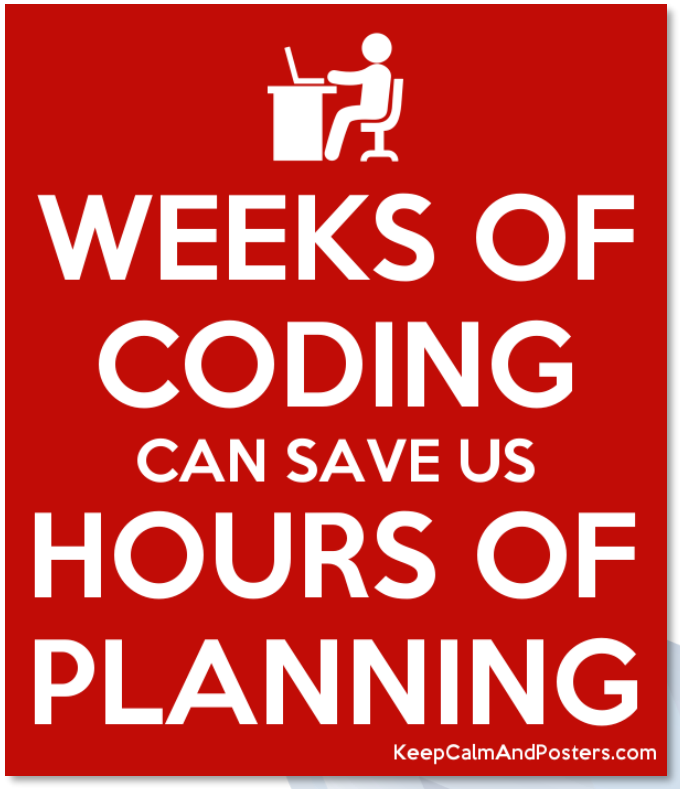
Why store data in scripts?

- Startup configs and credentials
- State persistence between executions
  - Compare current state to last state
- State persistence during (prolonged) execution
  - Be prepared for crash / early breakout
- State consolidation from multiple sources
- …and when juggling data is the script‘s purpose!

Not going to be talking about logging today ☺

Plan the data strategy

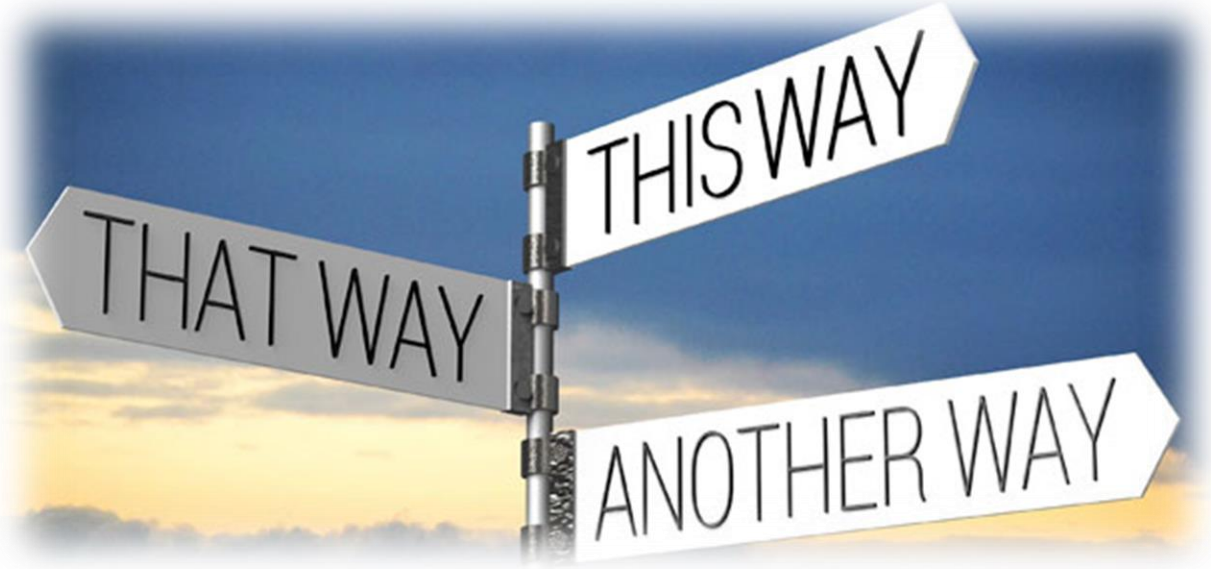
Choose wisely, refactoring is costly!



@cj_berlin

Mission Statement

- Right tools for the job
- As few dependencies as possible (portability)
- Cross-platform if possible (portability again)



- Performance *can* be an issue with large amounts

Planning questions

- Will multiple instances need simultaneous access?
  - Storage technology
  - Placement, connectivity and authentication
- Will carbon-based units be involved at any stage?
  - File and data formats – machine vs. human readability
  - Languages and locales
- What kind of data retention will apply?
  - Must the state outlive the source if it dies?

Storing Data: Flat Files

Flat Files in PowerShell: Objects

- PS Data File (.psd1)
  - import is nice
  - export is ugly
  - self-explanatory
- CLIXML
  - best object preservation
  - not very human-readable
- JSON
  - balance between human and machine readability
  - not available in legacy PS
- XML
  - industry standard for machine readability
  - not easy to work with

Flat Files in PowerShell: Scalars

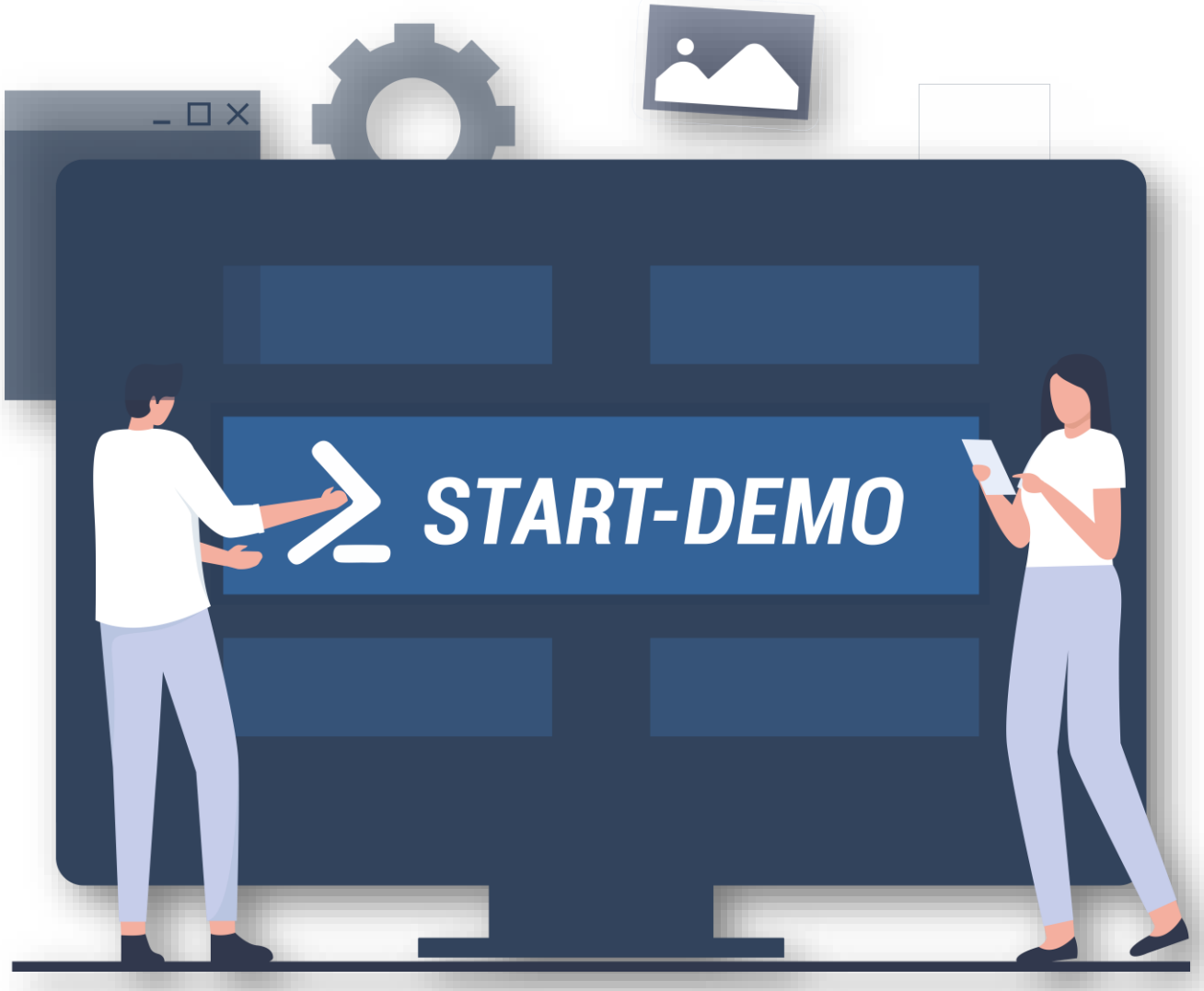
- CSV for tabular data
  - delimiter is variable…
  - …string qualifier is not
- Multi-line text files
  - no (real) random access
- Single-line text files
  - no random write access
- Binary files
  - no OOB support in PS
  - have to roll your own
  - compatibility to 3rd party
  - still no random write access

Common problems

- Encodings, especially in cross-platform scenarios
- Insert and update usually involve a complete rewrite
- Locking can get in the way of shared access

Demo

Flat Files



PSCONF.EU
VIENNA
20 - 23 June 2022

Storing Data: Databases

Querying is like reading, but way cooler!

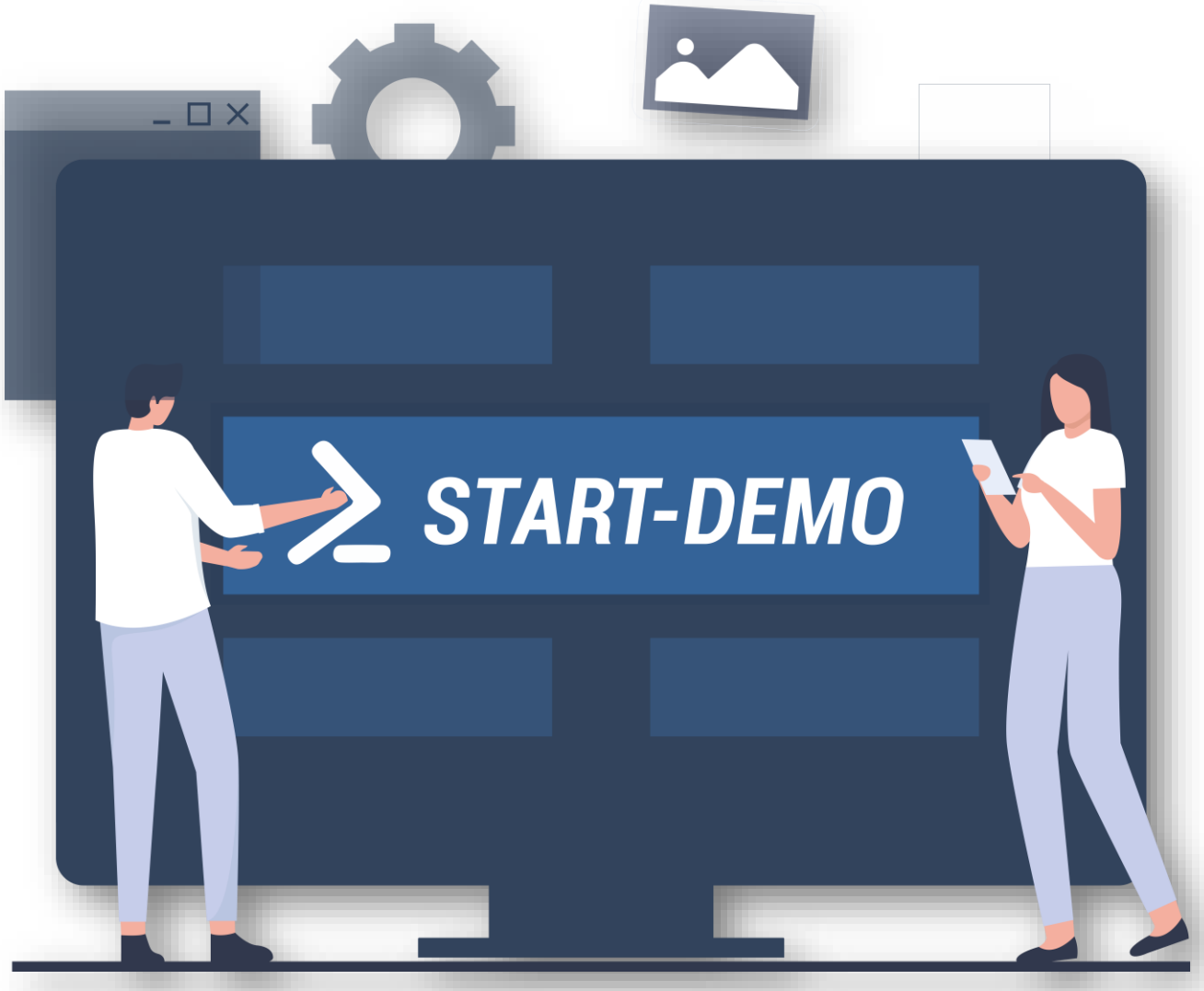
@cj_berlin

SQL

- SQL support is built into PowerShell (that is, .NET) in all versions and editions.
  - This way you can build a cross-platform centralised data store in SQL, once you figure out the authentication
  - Beware of older unpatched SQL servers when connecting from Linux or MacOS – TLS will get in the way!
- There are modules that make working with SQL more „PowerShell-y“ but then you have a dependency.

Demo

SQL



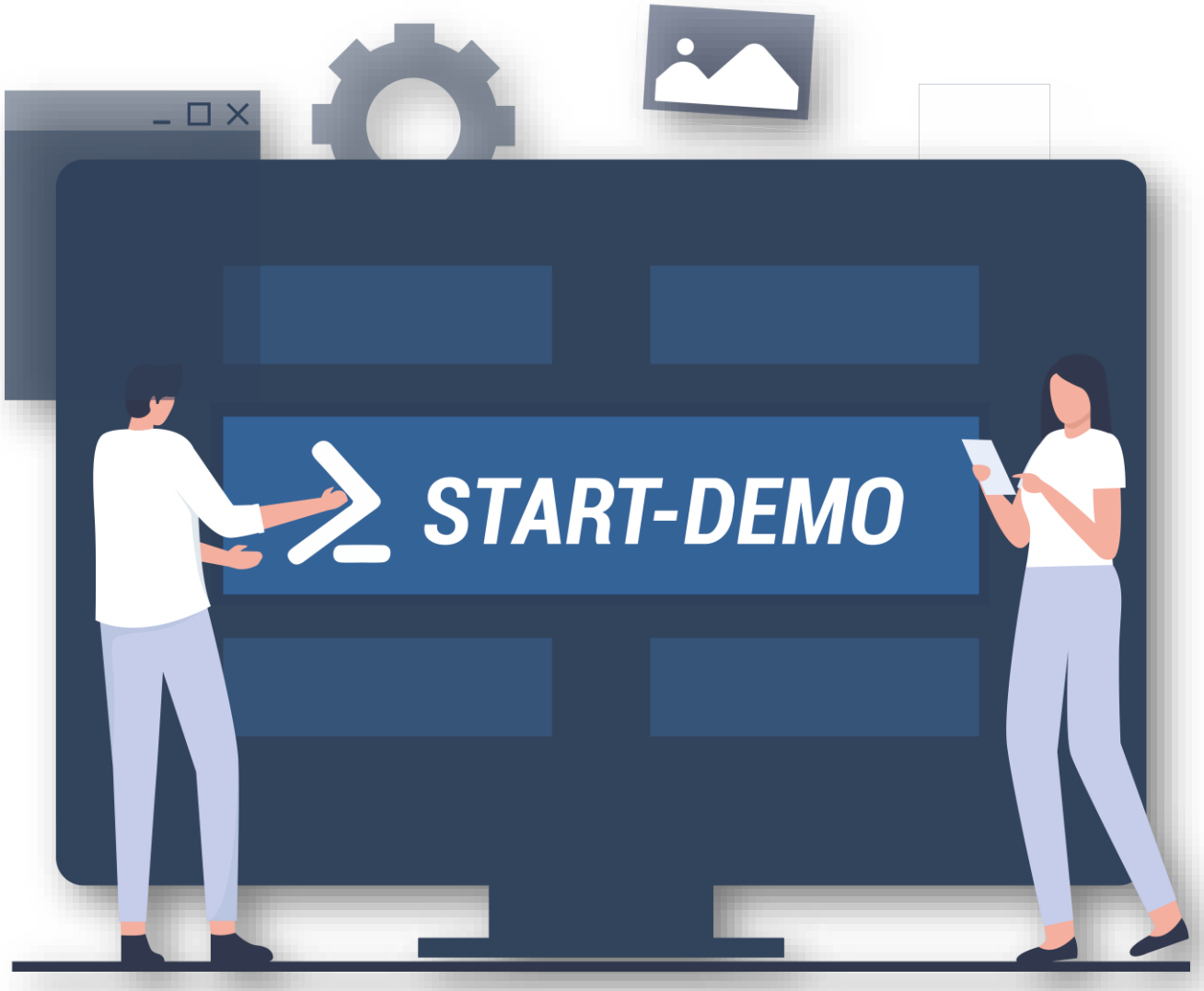
@cj_berlin

SQLite

- Not built-in, but:
  - free & open source
  - portable (as in installation-free, just copy the DLL) https://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki
  - cross-platform (the DLL is platform specific though)
  - supports in-memory DB: `"Data Source = :memory:"`
- Own SQL dialect, not 100% compatible to T-SQL
- Data types need getting used to

Demo

SQLite



Storing Data: Other places

Because sometimes we can, and sometimes we have to…

@cj_berlin

All over the world…

@cj_berlin

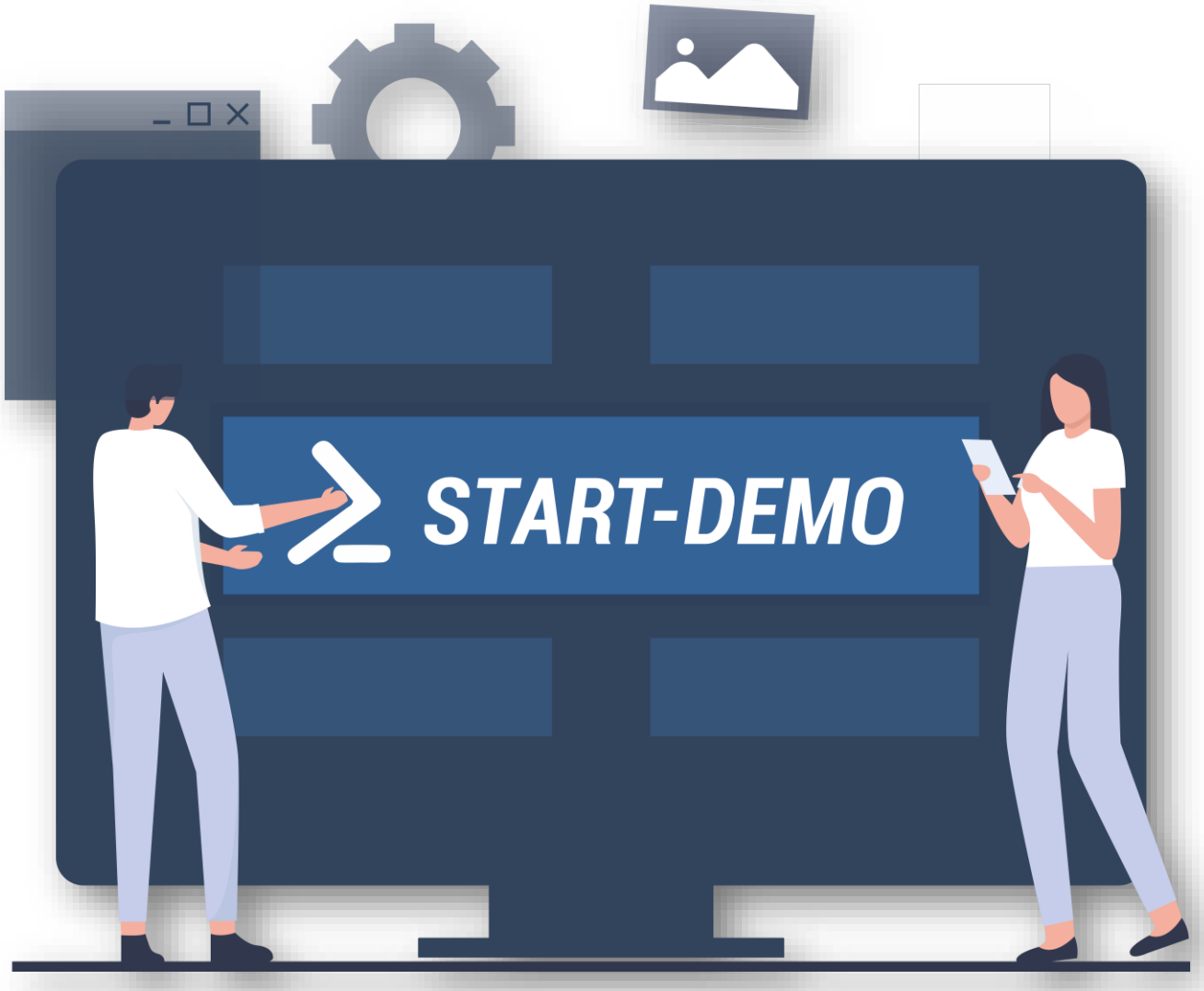
All over the world…

… why not use your existing Active Directory replication to distribute the data to all the locations?

- No need to worry about connectivity…
- …just do not use security-enabled objects
  - e.g. physicalLocation → description is a multi-valued attr
- Not quite cross-platform yet
- code usually looks rather messy

Demo

AD



@cj_berlin

If the connections are flaky…

… use a web service to GET, POST or PATCH you data.

- Azure Data Lake, Google Cloud Storage, …
- …or roll your own simple API

Wrappin‘ it up…

Right tools for the job…

- SQL for centralised storage of large amounts of data
  - cross-platform, authentication can be a challenge
- SQLite for ultra-fast local relational data storage
  - files are cross-platform as well, in-memory possible
- CLIXML for best object persistence
- AD for built-in replication
- REST API for best resilience and connectivity

Bonus: SQL pointers

- Use argument preparation to avoid SQL injection
- One sqlCommand object does not allow for simultaneous read and write
- Be aware that `[DBNull]:Value -ne $null` → **DEMO**
- Try to stick with integrated authentication and use least privilege (e.g. blackhole permissions)

Bonus: SQLite pointers

- Using a shared SQLite DB on a file share is not at all encouraged by the maintainers: https://www.sqlite.org/draft/useovernet.html
- Always close the DB connection and invoke `[gc]::Collect()` before exiting!
- Upsert works differently from SQL: `INSERT OR REPLACE`
- SQLite in memory can transcend runspaces! → **DEMO**

Q&A

Thank you!

https://github.com/it-pro-berlin-de/esm-psconfeu22

@cj_berlin