DEVELOPING AND TESTING CROSS PLATFORM COMPATIBLE MODULES WITH POWERSHELL CORE

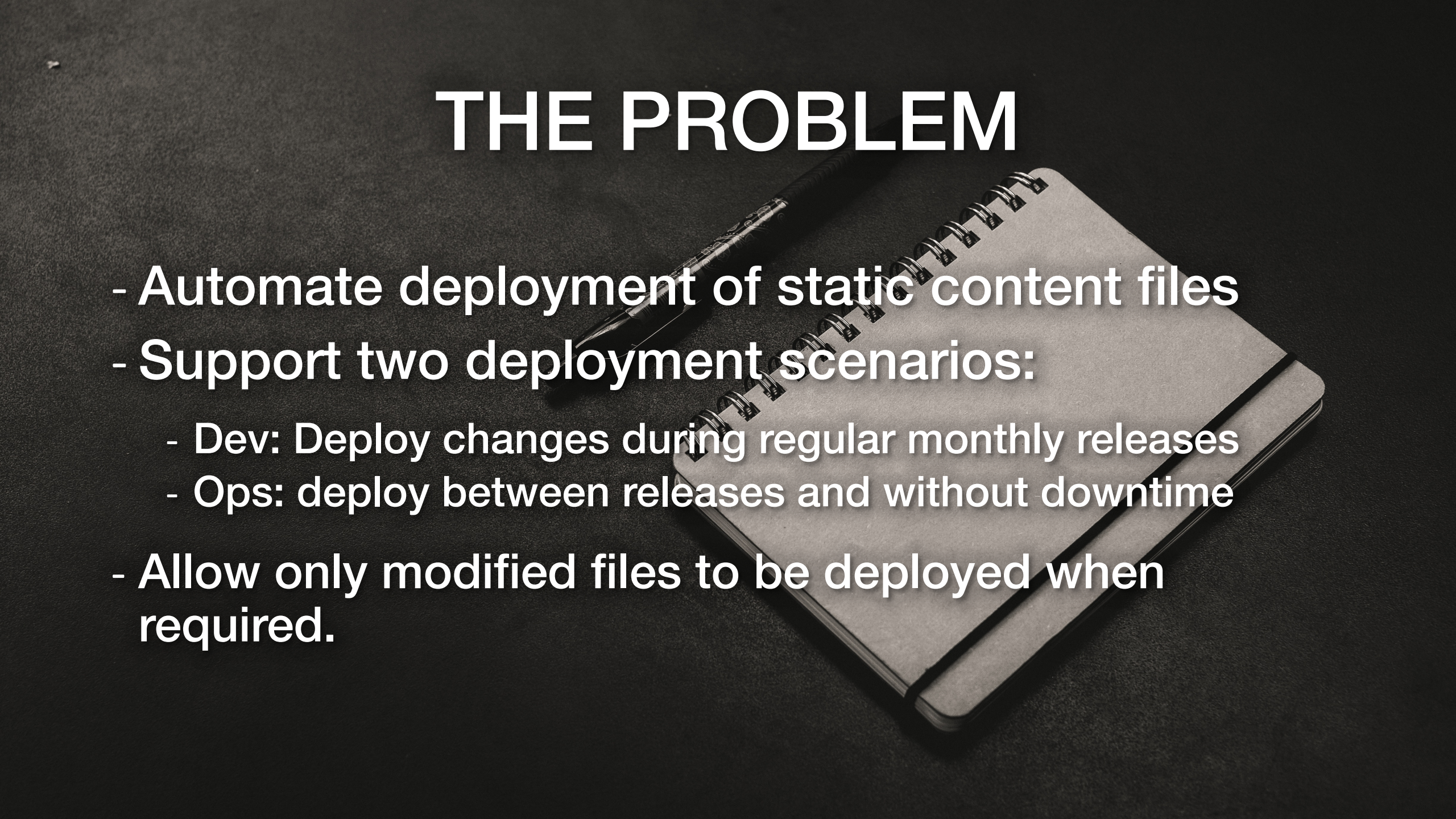
MARK WRAGG

DevOps Engineer, XE.com

@markwragg

https://wragg.io

THE PROBLEM



- Automate deployment of static content files
- Support two deployment scenarios:
  - Dev: Deploy changes during regular monthly releases
  - Ops: deploy between releases and without downtime
- Allow only modified files to be deployed when required.

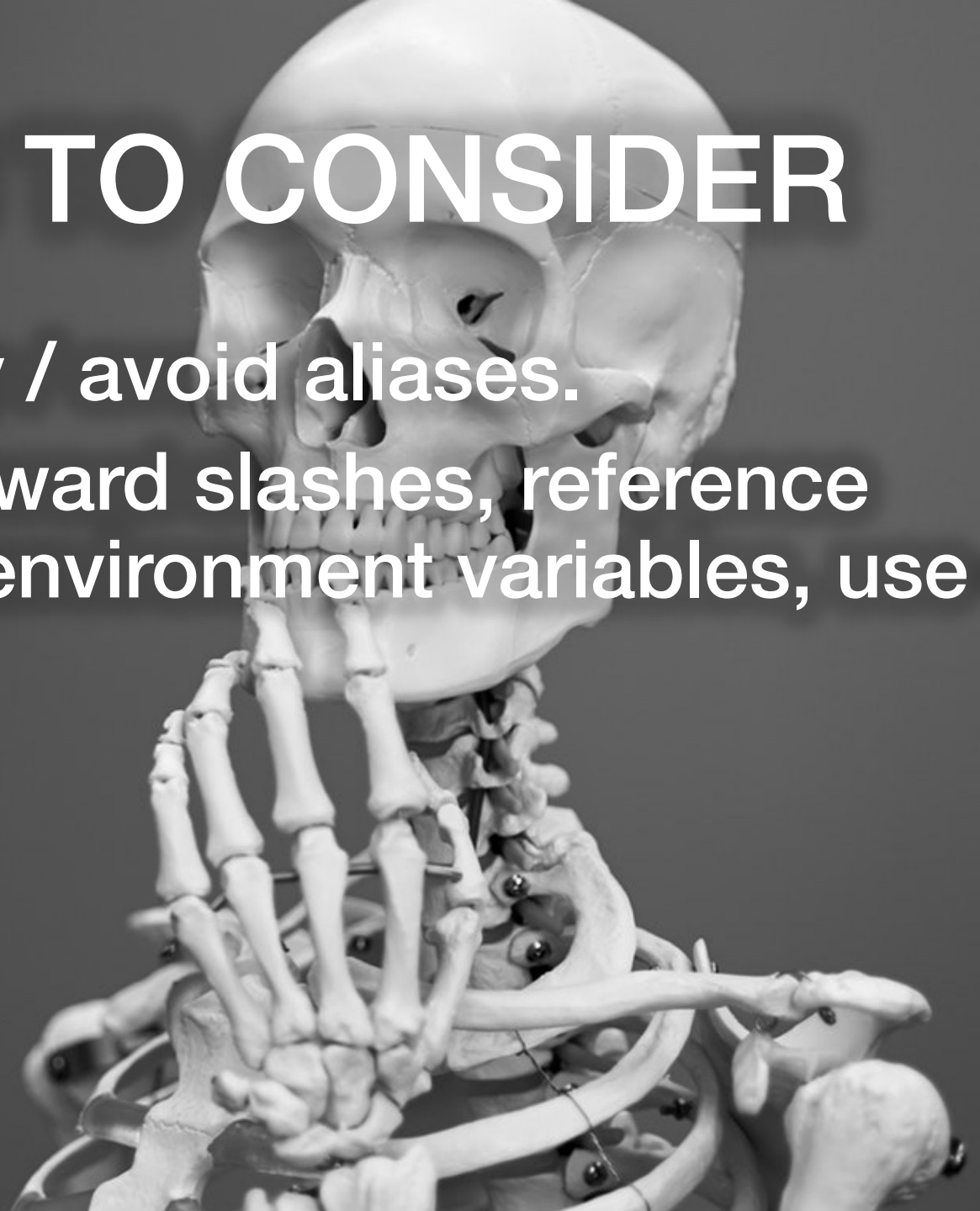
THE SOLUTION

POWERSHELL CORE

- New, fast
- Open-source, cross-platform
- Not all cmdlets available / modules compatible

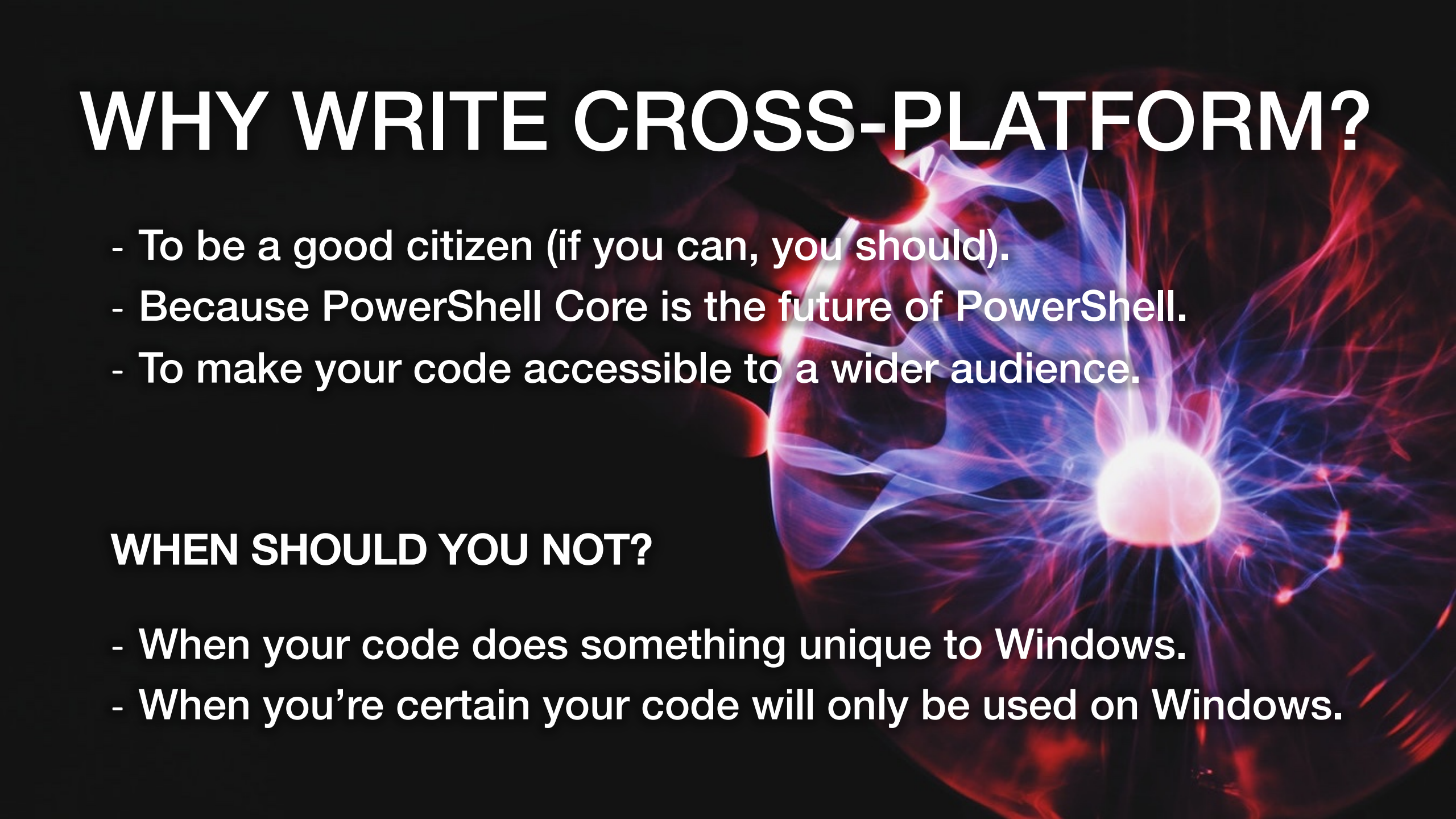
THINGS TO CONSIDER

- Cmdlet availability / avoid aliases.
- File paths: use forward slashes, reference system paths via environment variables, use –path cmdlets.
- Encode as UTF8.



WHY WRITE CROSS-PLATFORM?

- To be a good citizen (if you can, you should).
- Because PowerShell Core is the future of PowerShell.
- To make your code accessible to a wider audience.

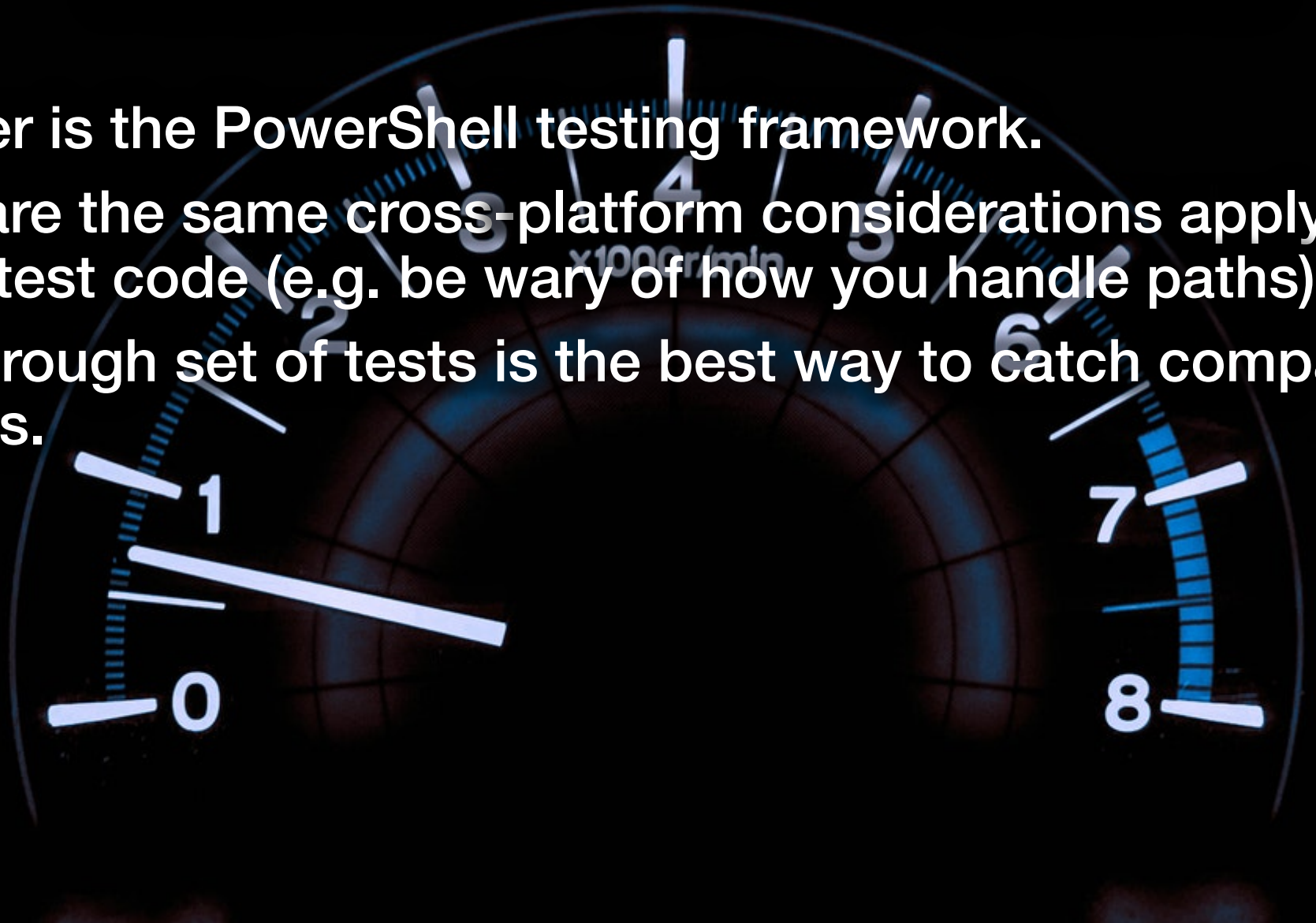


WHEN SHOULD YOU NOT?

- When your code does something unique to Windows.
- When you're certain your code will only be used on Windows.

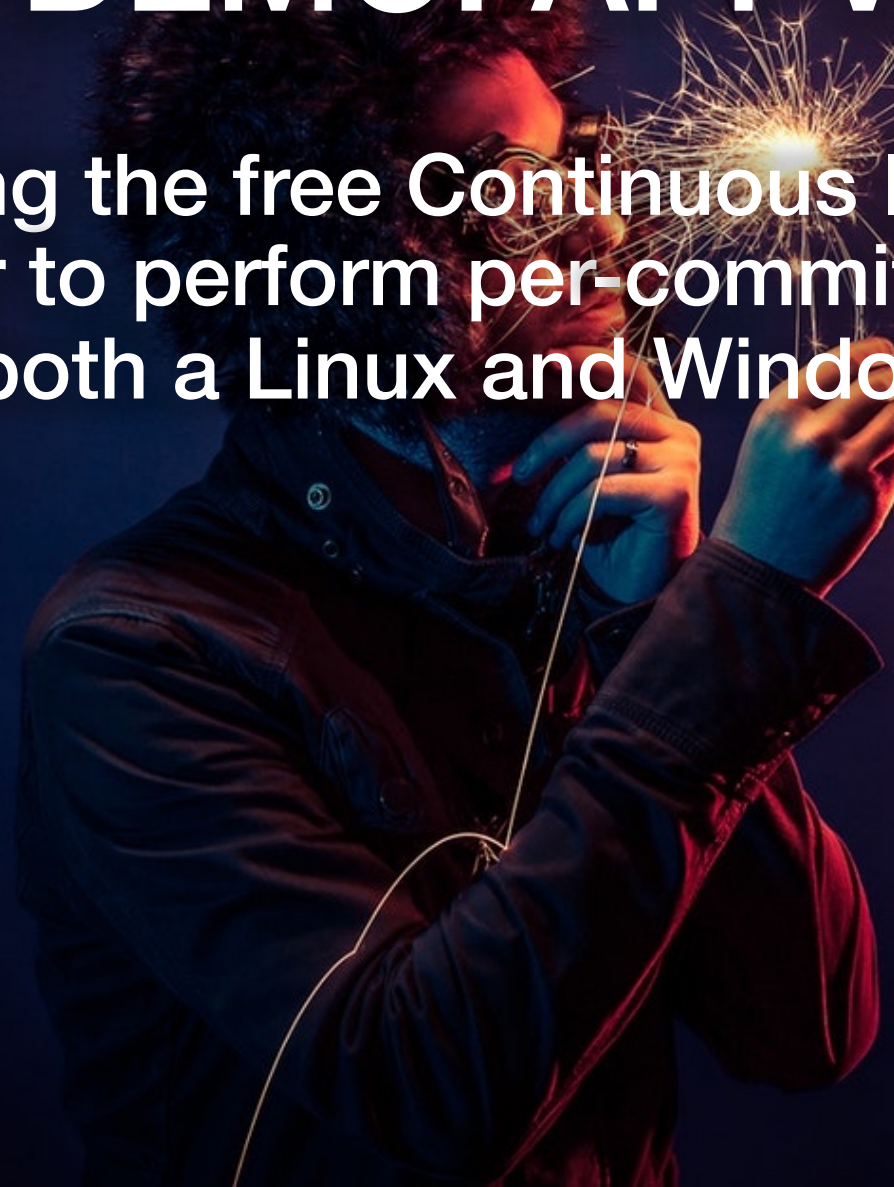
TESTING YOUR CODE

- Pester is the PowerShell testing framework.
- Beware the same cross-platform considerations apply to your test code (e.g. be wary of how you handle paths).
- A thorough set of tests is the best way to catch compatibility issues.



DEMO: APPVEYOR

- Leveraging the free Continuous Integration service AppVeyor to perform per-commit Pester testing of code on both a Linux and Windows VM.



SUMMARY

- Review your existing code for opportunities to make it cross-platform (or confirm if it already is!).
- You can ensure your code remains cross-platform compatible via automated testing.
- Tag your PS Core compatible code as such in the PS Gallery.
- Links:
  - https://www.powershellgallery.com/packages/HashCopy/
  - https://github.com/markwragg/PowerShell-HashCopy
  - https://www.appveyor.com/

Twitter: @markwragg

Blog: https://wragg.io