```
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'
```

'wind\x6f\x77["\x6

\x65E\x6cemen\x74'

Hex
0x## - Numbers
\x## - Printable Chars

\x74

'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

What is that?

Hex
0x## - Numbers
\x## - Printable Chars

\x74

```
> [Convert]::ToInt16("74",16)
116
```

'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

What is that?

Hex

0x## - Numbers

\x## - Printable Chars

\x74

```
> [Convert]::ToInt16("74",16)
```

116

```
> [Char](116)
```

t

```
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'
```

```
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'
```

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches
```

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){
```

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){

   # Get just the value without the hex prefix
   $charVal    = $char.Substring(2,2)
```

How can we make sense of it?

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){

   # Get just the value without the hex prefix
   $charVal    = $char.Substring(2,2)

   # Convert the value from base 16 to base 10
   $asciiVal   = [Convert]::ToInt16($charVal,16)
```

How can we make sense of it?

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){

   # Get just the value without the hex prefix
   $charVal    = $char.Substring(2,2)

   # Convert the value from base 16 to base 10
   $asciiVal   = [Convert]::ToInt16($charVal,16)

   # Convert the base 10 int to the ASCII char
   $asciiChar  = [Char]($asciiVal)
```

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){

   # Get just the value without the hex prefix
   $charVal    = $char.Substring(2,2)

   # Convert the value from base 16 to base 10
   $asciiVal   = [Convert]::ToInt16($charVal,16)

   # Convert the base 10 int to the ASCII char
   $asciiChar  = [Char]($asciiVal)

   # Replace the hex char with the ASCII char
   $cleanstring = $cleanstring.Replace($char,$asciiChar)
}
```

```powershell
# Our hex encoded string
$cleanstring = $string =
'wind\x6f\x77["\x64oc\x75\x6den\x74"\x5d\x2ec\x72\x65a\x74\x65E\x6cemen\x74'

# Use regex to pull out the literal \x## matches as a capture group
$hexChars   = $string | Select-String -Pattern "([.\\][.x].{2})" -AllMatches

# Loop through each match from our regex
foreach($char in $hexChars.Matches.Value){

   # Get just the value without the hex prefix
   $charVal    = $char.Substring(2,2)

   # Convert the value from base 16 to base 10
   $asciiVal   = [Convert]::ToInt16($charVal,16)

   # Convert the base 10 int to the ASCII char
   $asciiChar  = [Char]($asciiVal)

   # Replace the hex char with the ASCII char
   $cleanstring = $cleanstring.Replace($char,$asciiChar)
}

> $cleanstring
window["document"].createElement
```

Can't we just use AI?

But no, PowerShell is good!

We need to balance the risks & benefits

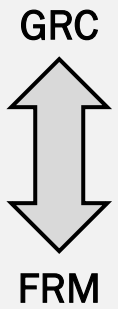
What is risk?

Probability

What is risk?

How do we manage risk?

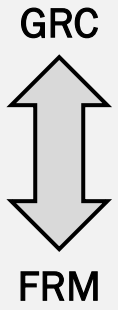
Risk Management Frameworks (**RMF**)



GRC: Governance, Risk, Compliance
FRM: Financial Risk Management

How do we manage risk?

Risk Management Frameworks (**RMF**)



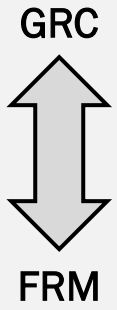
ISO – General

COSO – Financial Centric

ISO: International Organization for Standardization
COSO: Committee of Sponsoring Organizations
GRC: Governance, Risk, Compliance
FRM: Financial Risk Management

How do we manage risk?

Risk Management Frameworks (**RMF**)



FAIR – Quantitative Focus

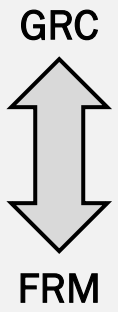
ISO – General

COSO – Financial Centric

ISO: International Organization for Standardization
COSO: Committee of Sponsoring Organizations
GRC: Governance, Risk, Compliance
FRM: Financial Risk Management

How do we manage risk?

Risk Management Frameworks (**RMF**)



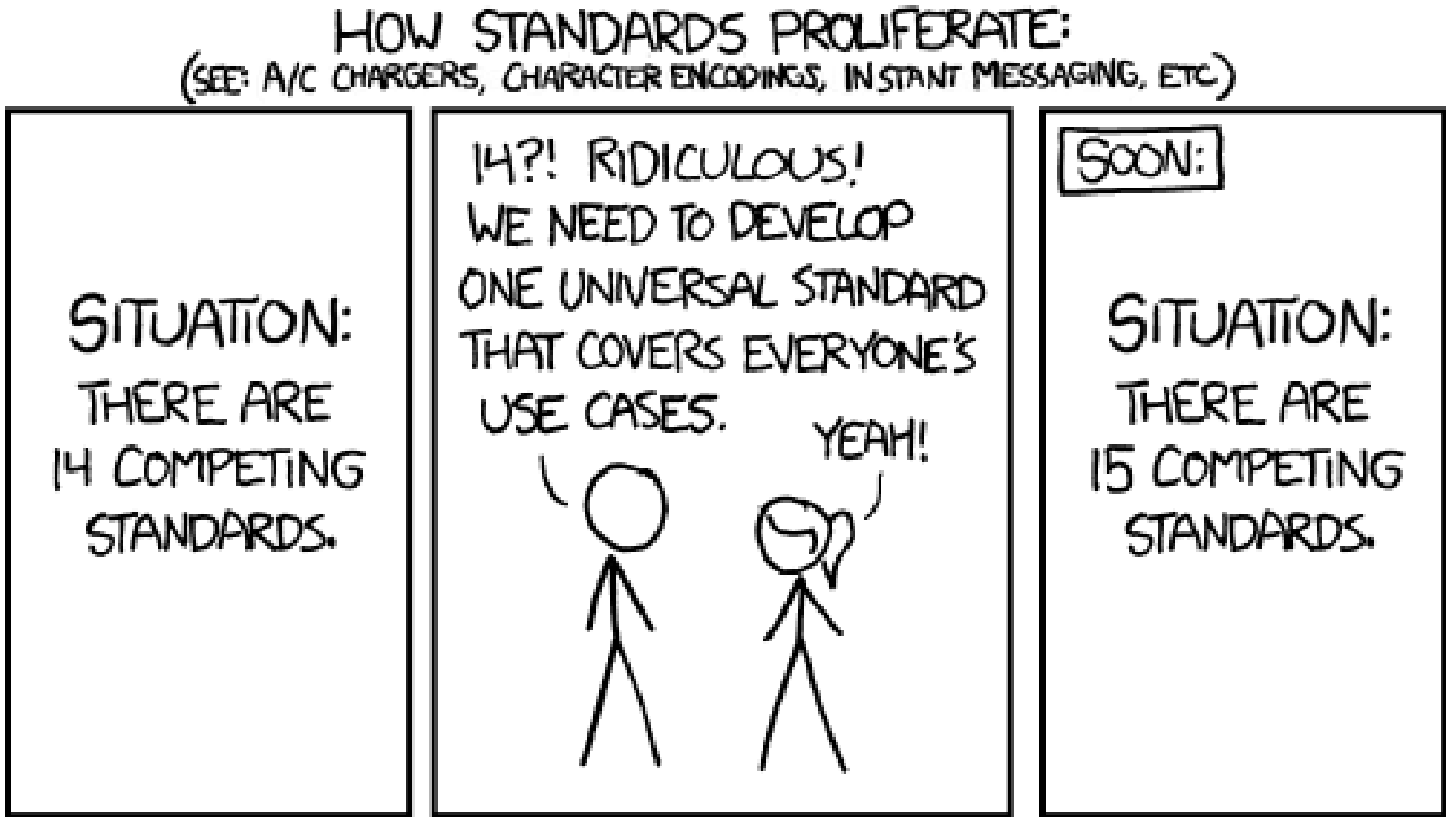
NIST/ITIL/COBIT – IT Centric

FAIR – Quantitative Focus

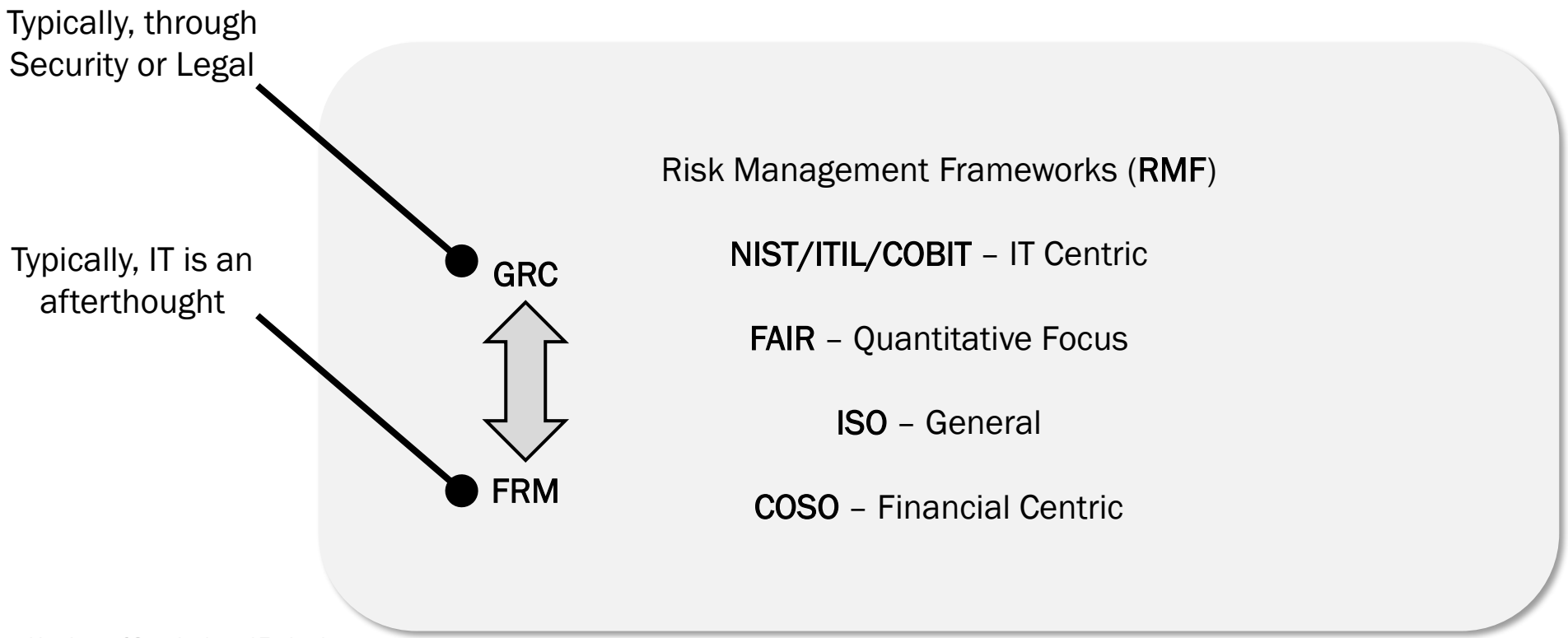
ISO – General

COSO – Financial Centric

NIST: National Institute of Standards and Technology
ITIL: Information Technology Infrastructure Library
COBIT: Control Objectives for Information and Related Technologies
FAIR: Factor Analysis of Information Risk
ISO: International Organization for Standardization
COSO: Committee of Sponsoring Organizations
GRC: Governance, Risk, Compliance
FRM: Financial Risk Management

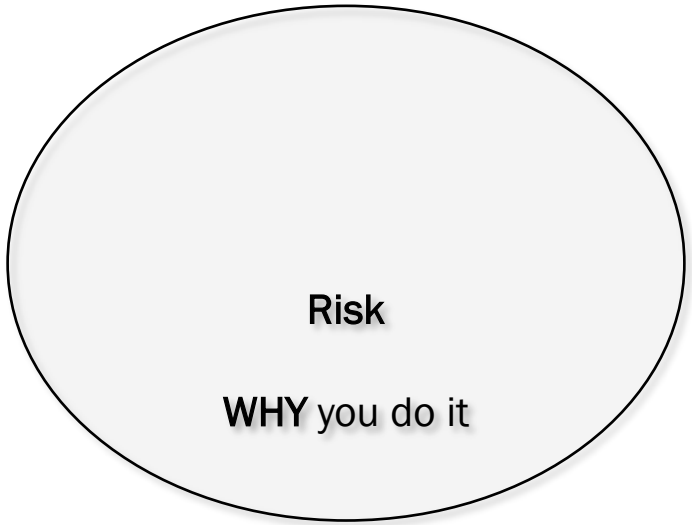

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

How do we manage risk?



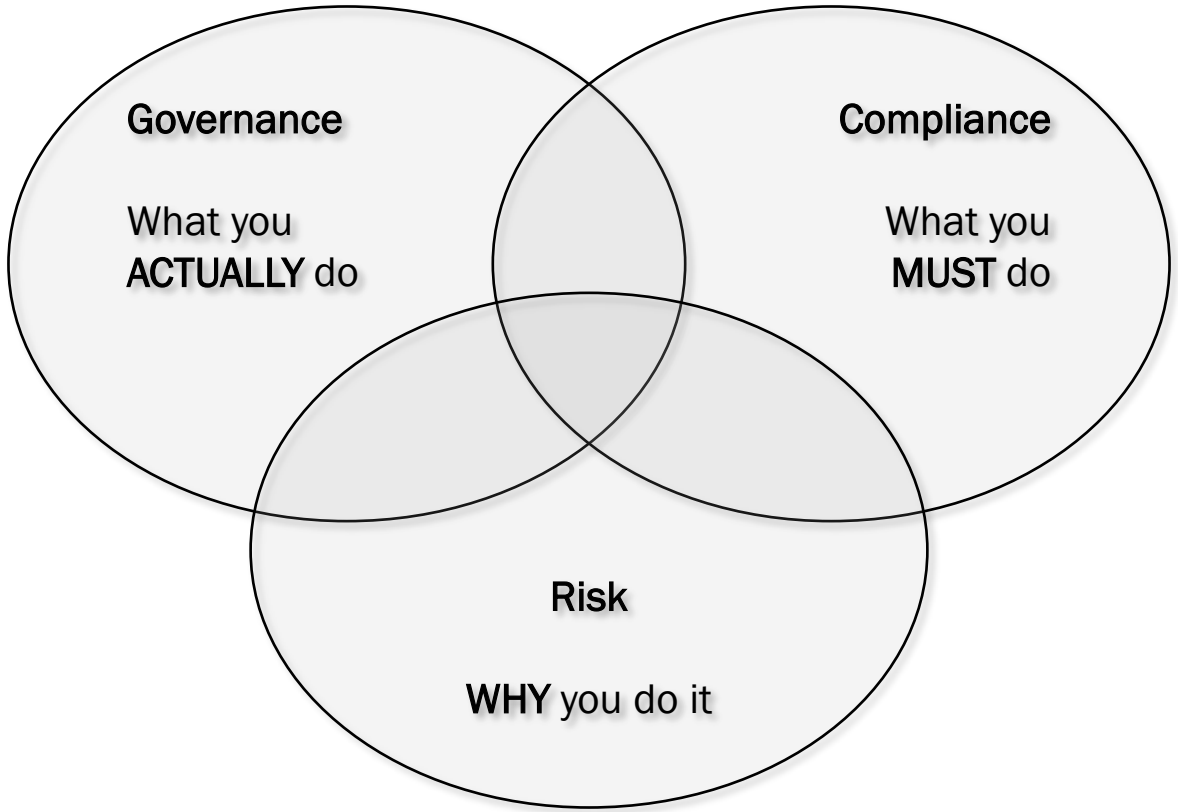
NIST: National Institute of Standards and Technology
ITIL: Information Technology Infrastructure Library
COBIT: Control Objectives for Information and Related Technologies
FAIR: Factor Analysis of Information Risk
ISO: International Organization for Standardization
COSO: Committee of Sponsoring Organizations
GRC: Governance, Risk, Compliance
FRM: Financial Risk Management

What is GRC then?

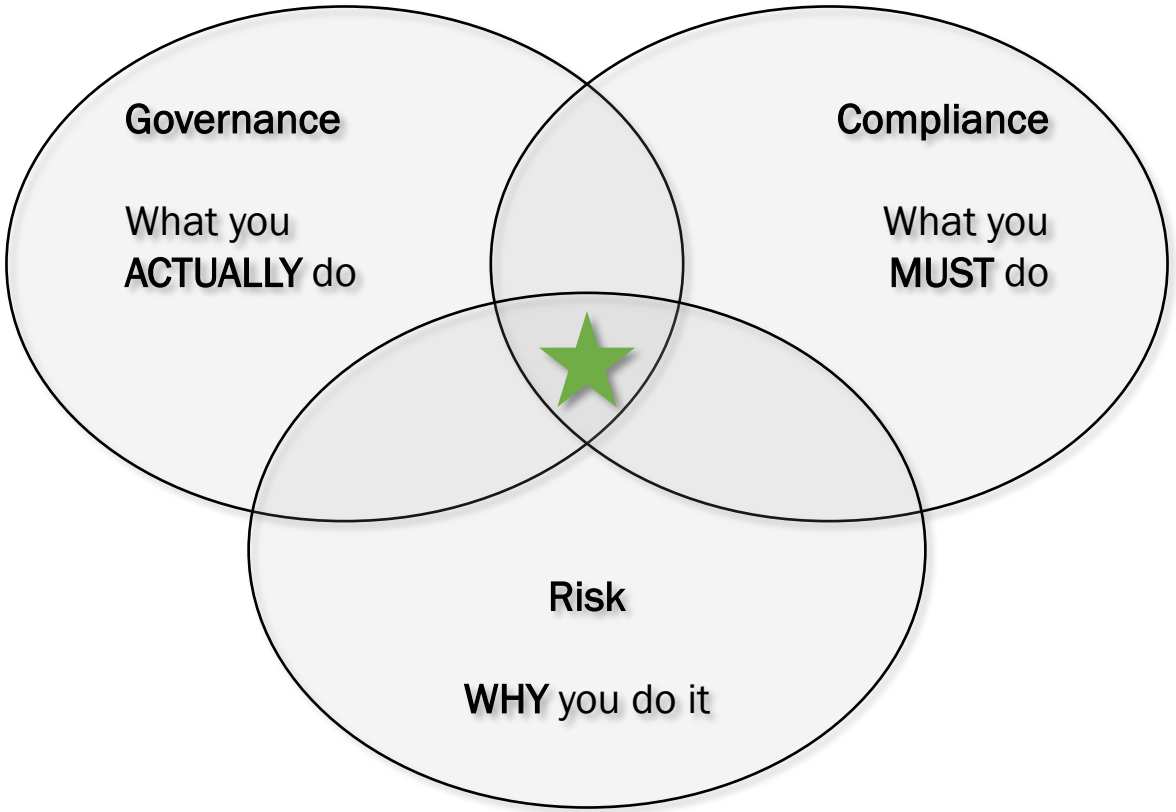


What is GRC then?

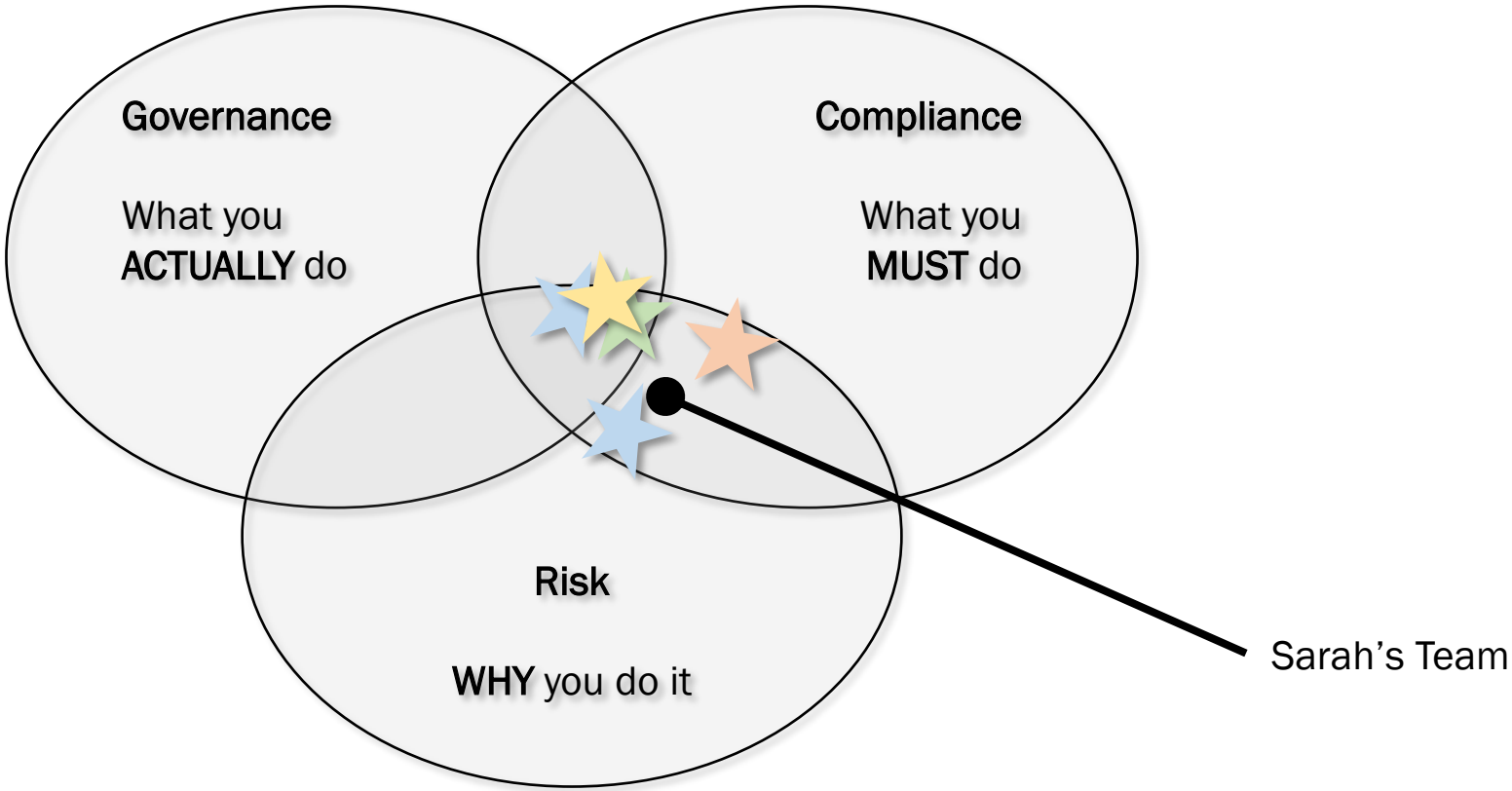
What is GRC then?



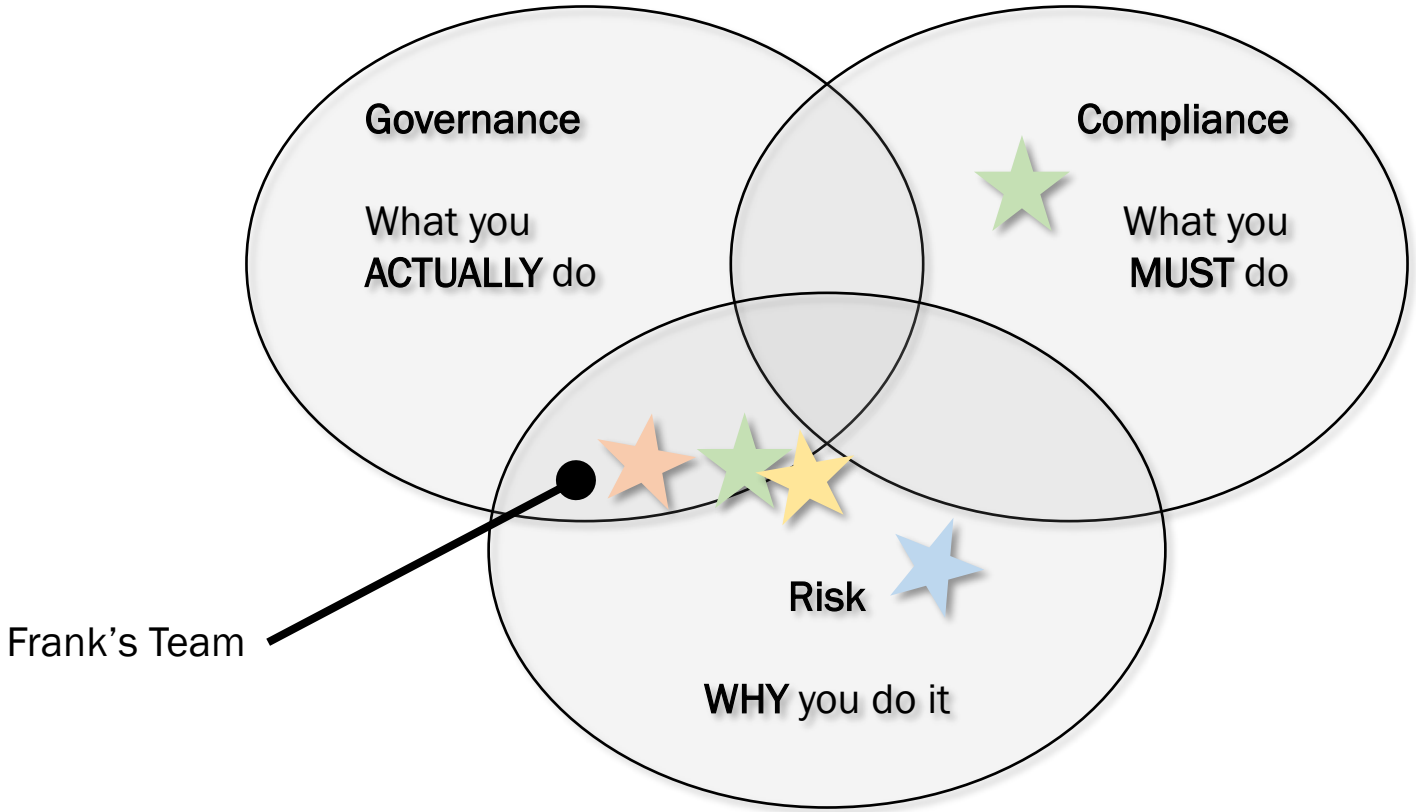
Unfortunately, it isn't a "We are here" representation



It is more like this…



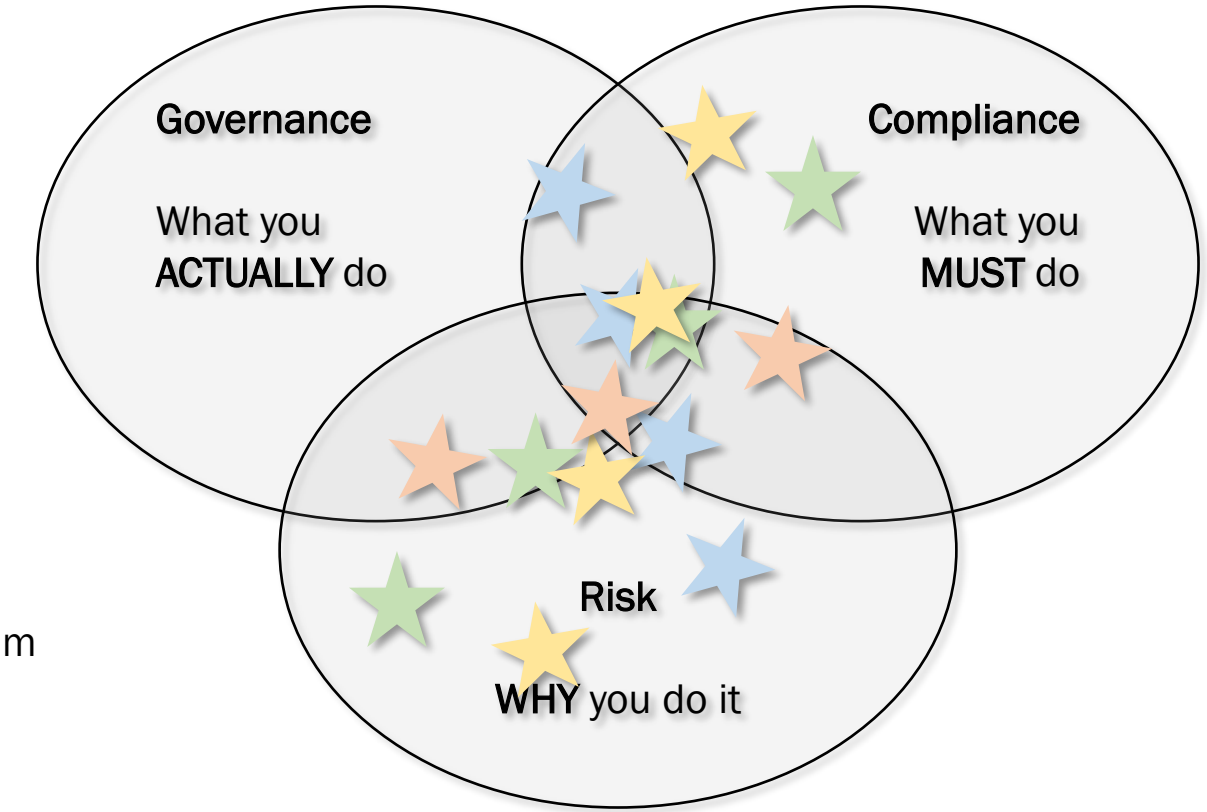
It is more like this…



It is more like this…

It is more like this…

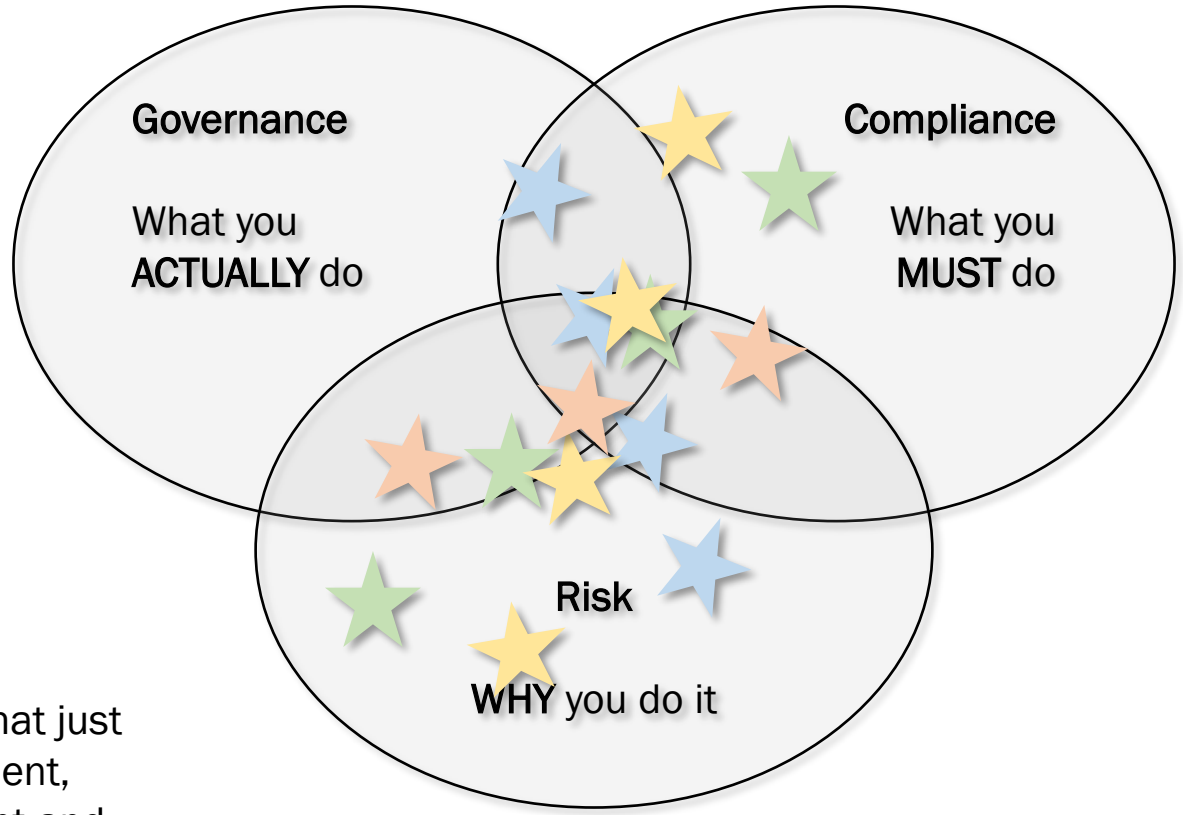
Mike's Team



Frank's Team

Sarah's Team

It is more like this…



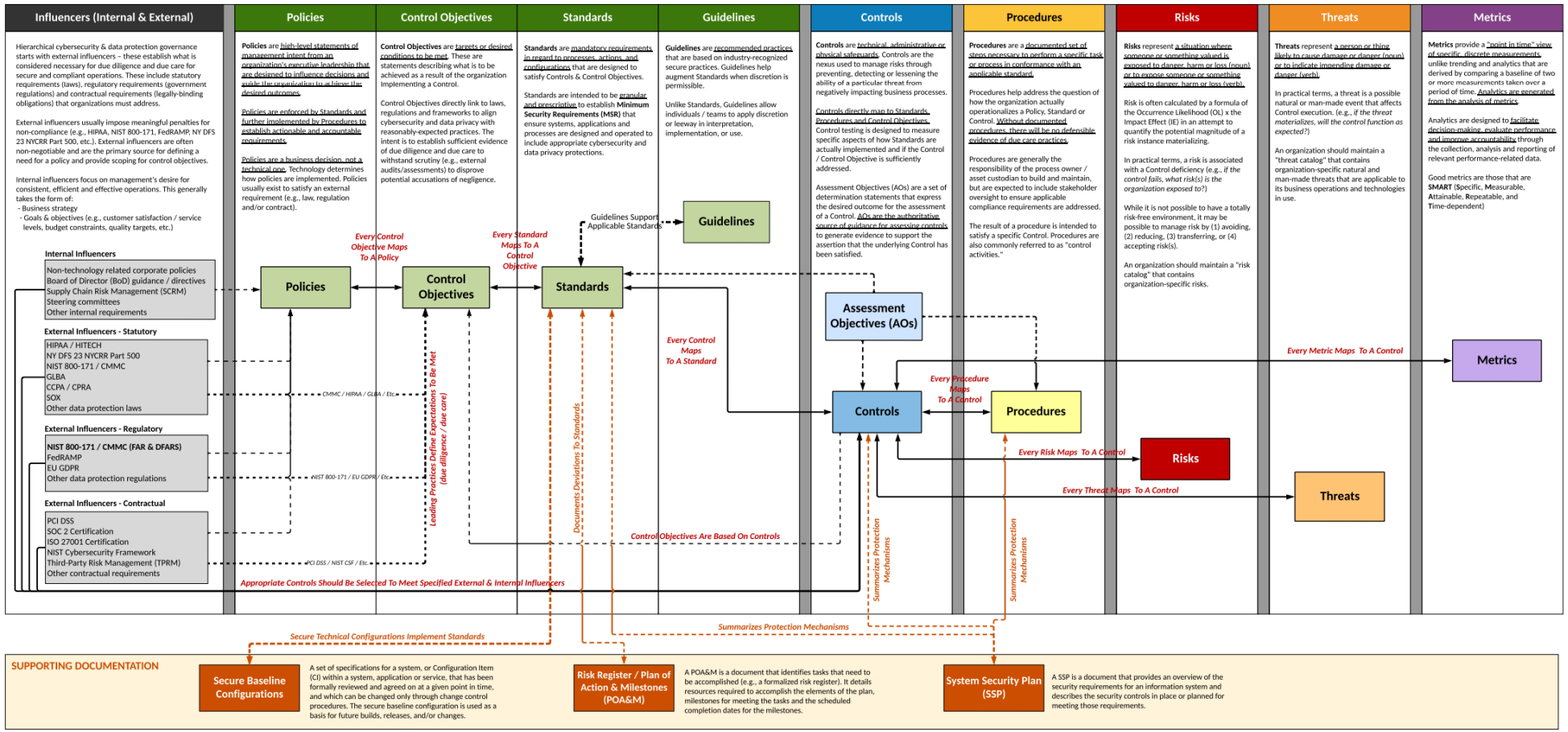
Some random group that just bypassed procurement, bought new equipment and software, and is mining Bitcoin to fund their projects

How do we actually succeed then?

How do we actually succeed then?

ALWAYS LEADING

COMPLIANCE FORGE



SENTINEL

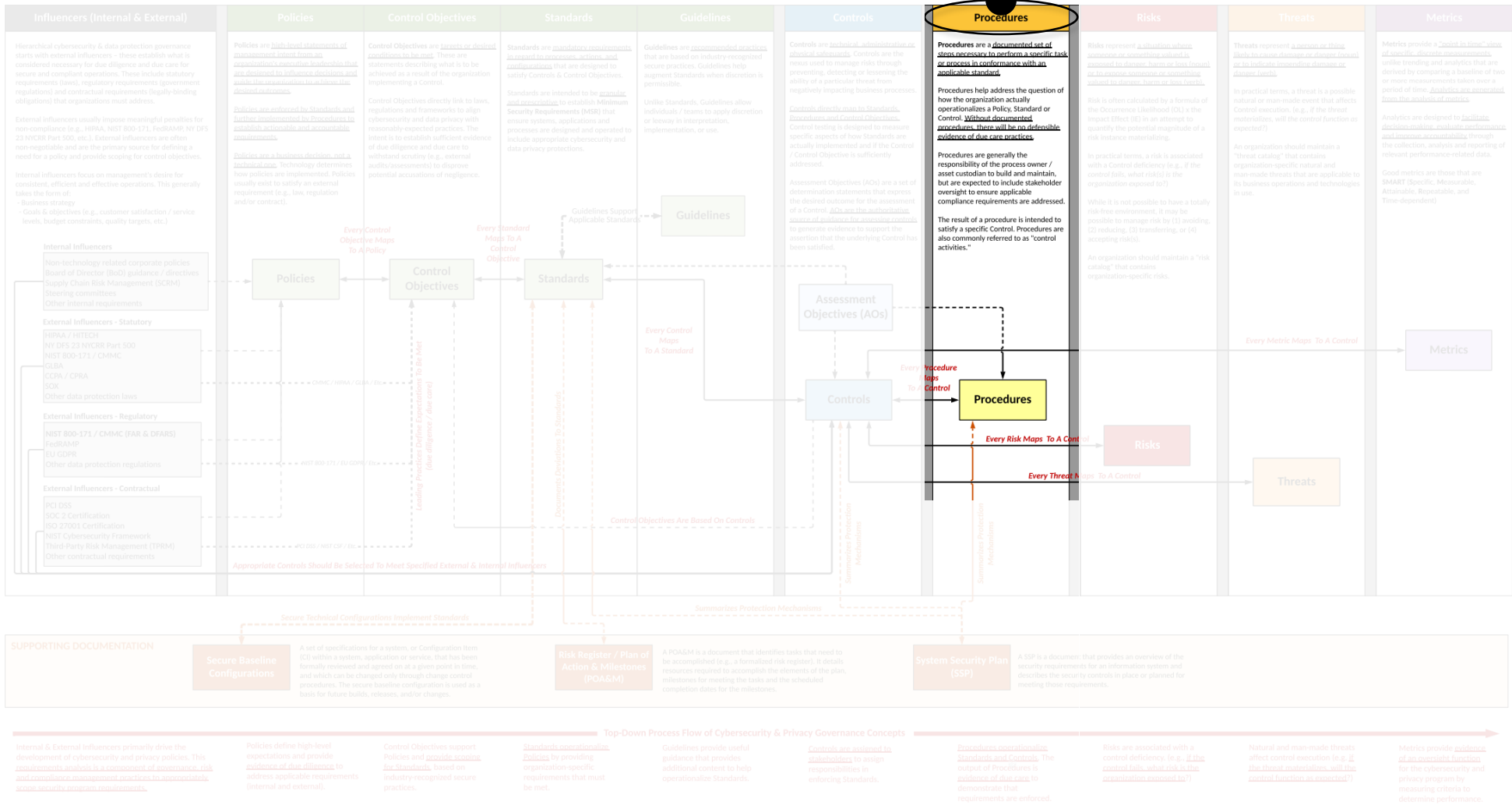
Linkedin.com/in/mikesoule/

Research Triangle Park PowerShell User Group

How do we actually succeed then?

COMPLIANCE FORGE

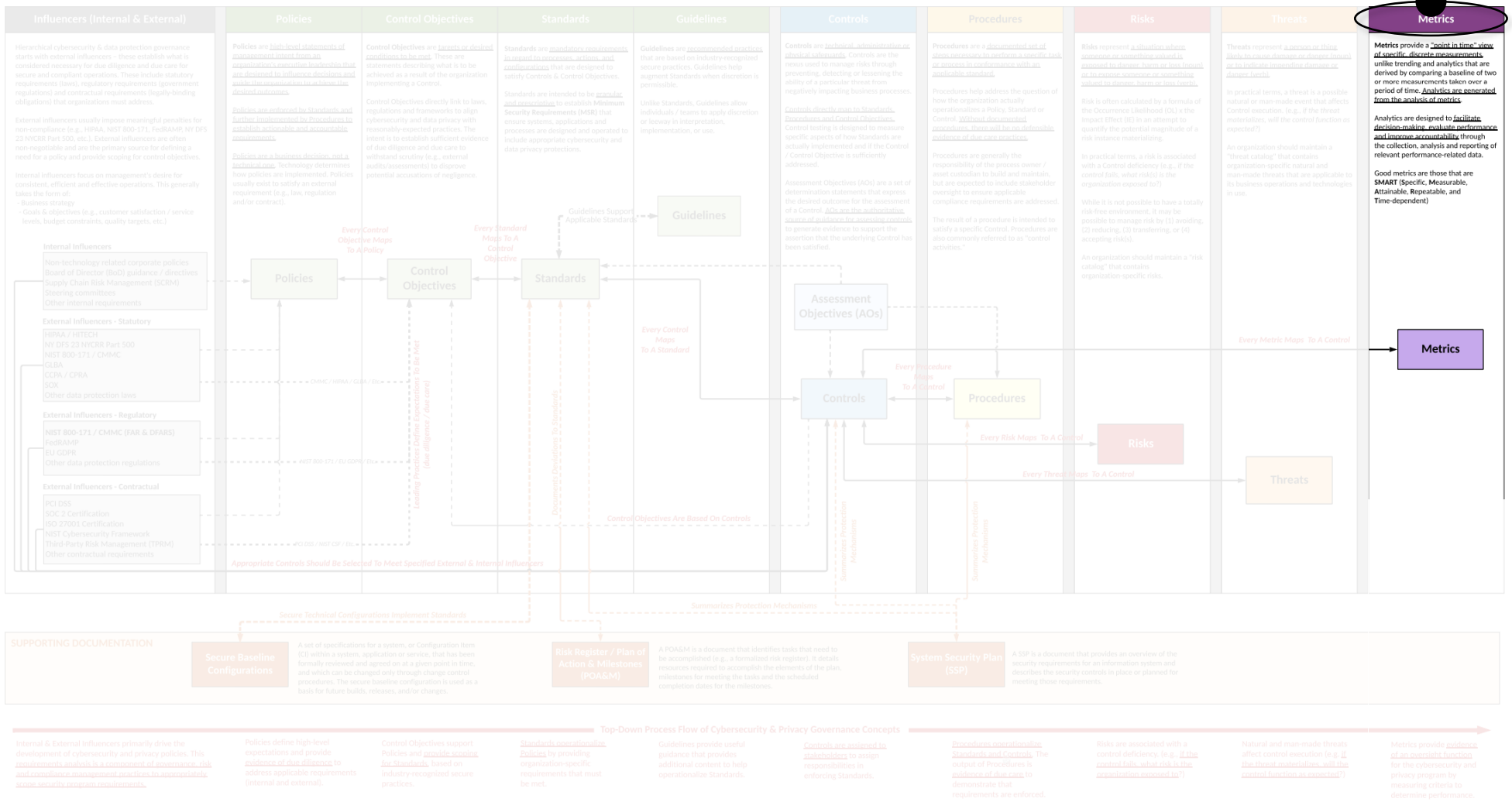
Most IT Ops teams do this, but only occasionally well…



How do we actually succeed then?

COMPLIANCE FORGE

Sometimes you'll even be doing this...

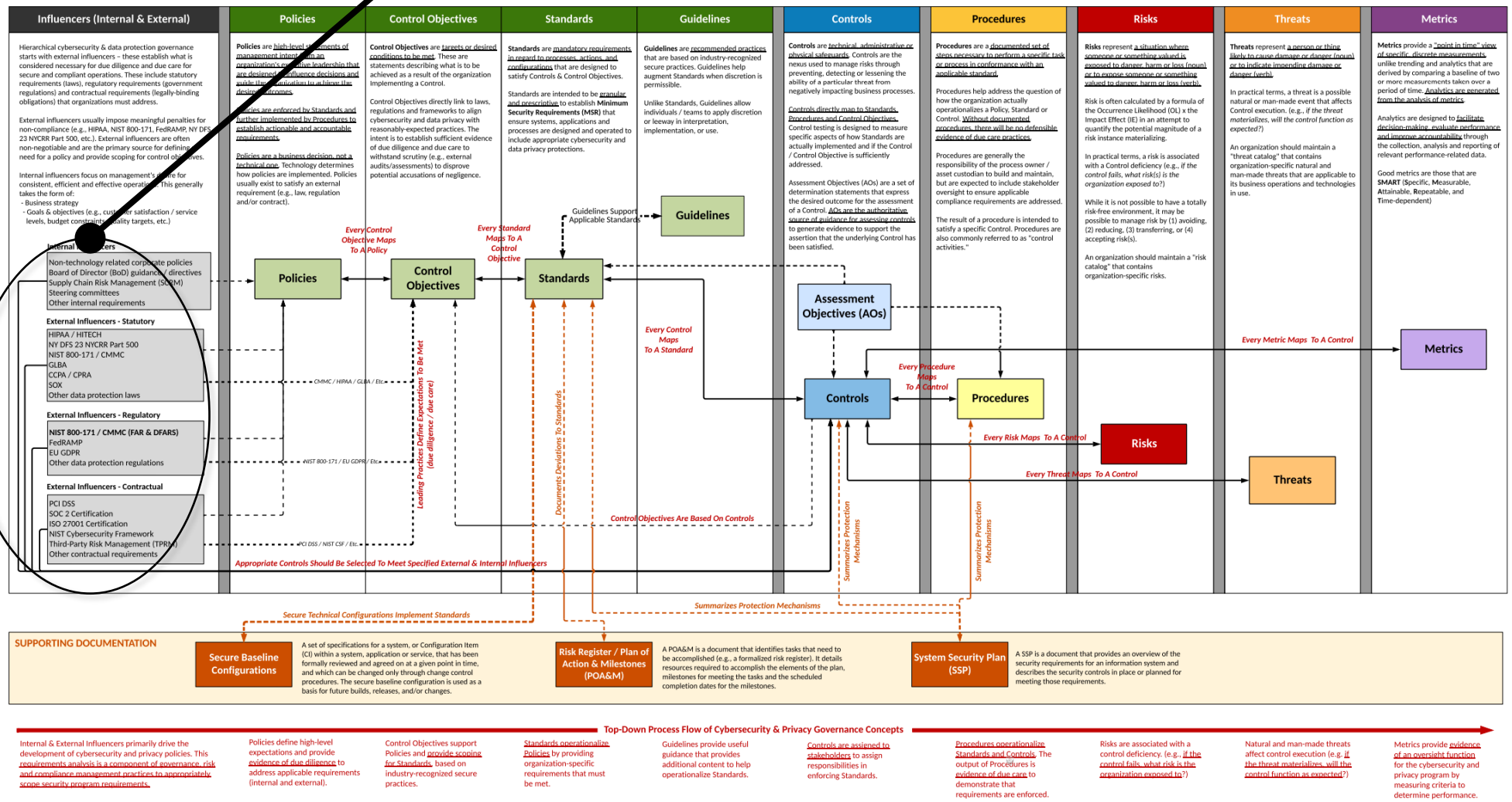


How do we actually succeed then?

ALWAYS LEADING

COMPLIANCE FORGE

Operations, Security, & Risk all need to align

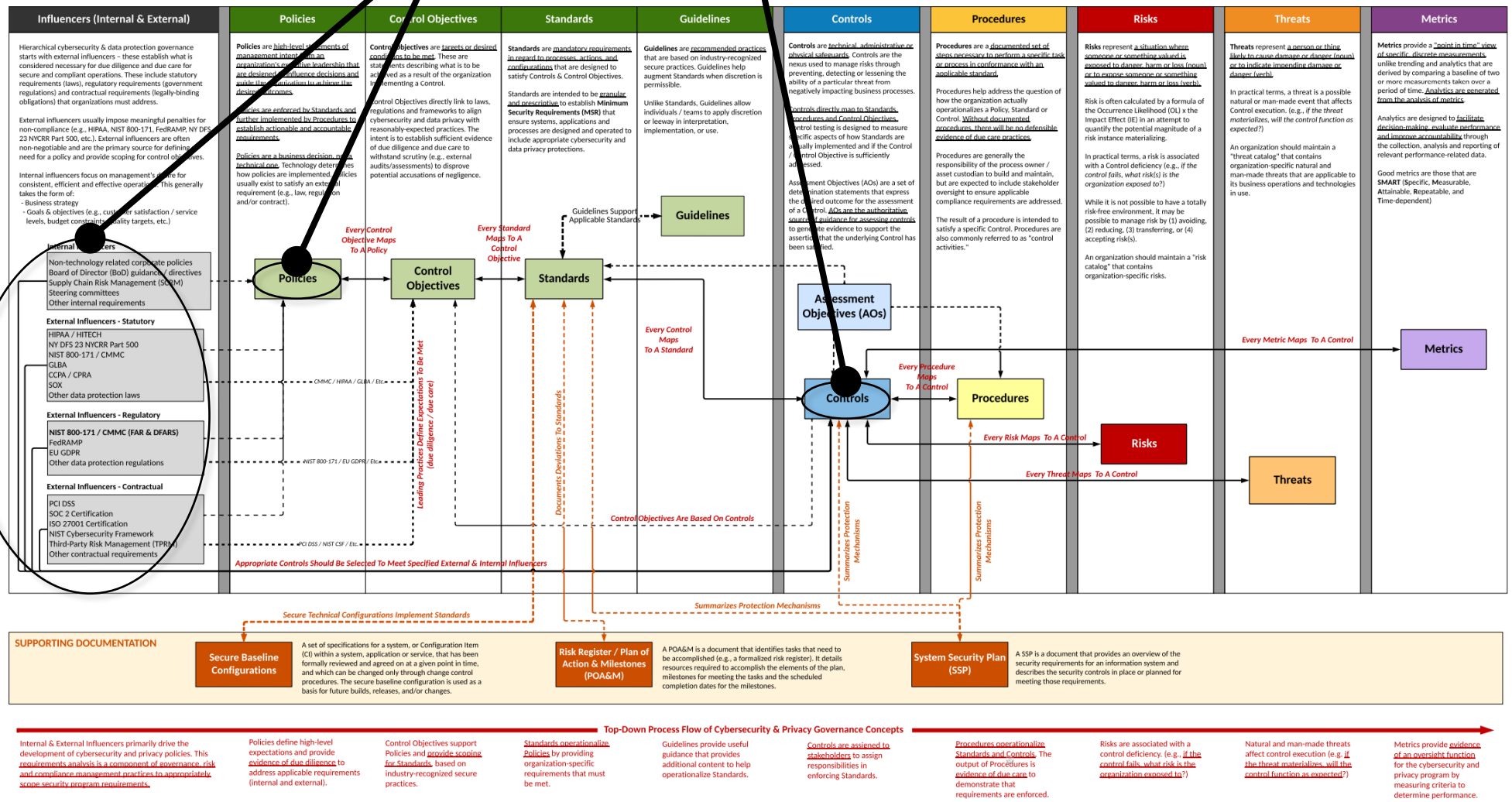


How do we actually succeed then?

ALWAYS LEADING

COMPLIANCE FORGE

Operations, Security, & Risk all need to align

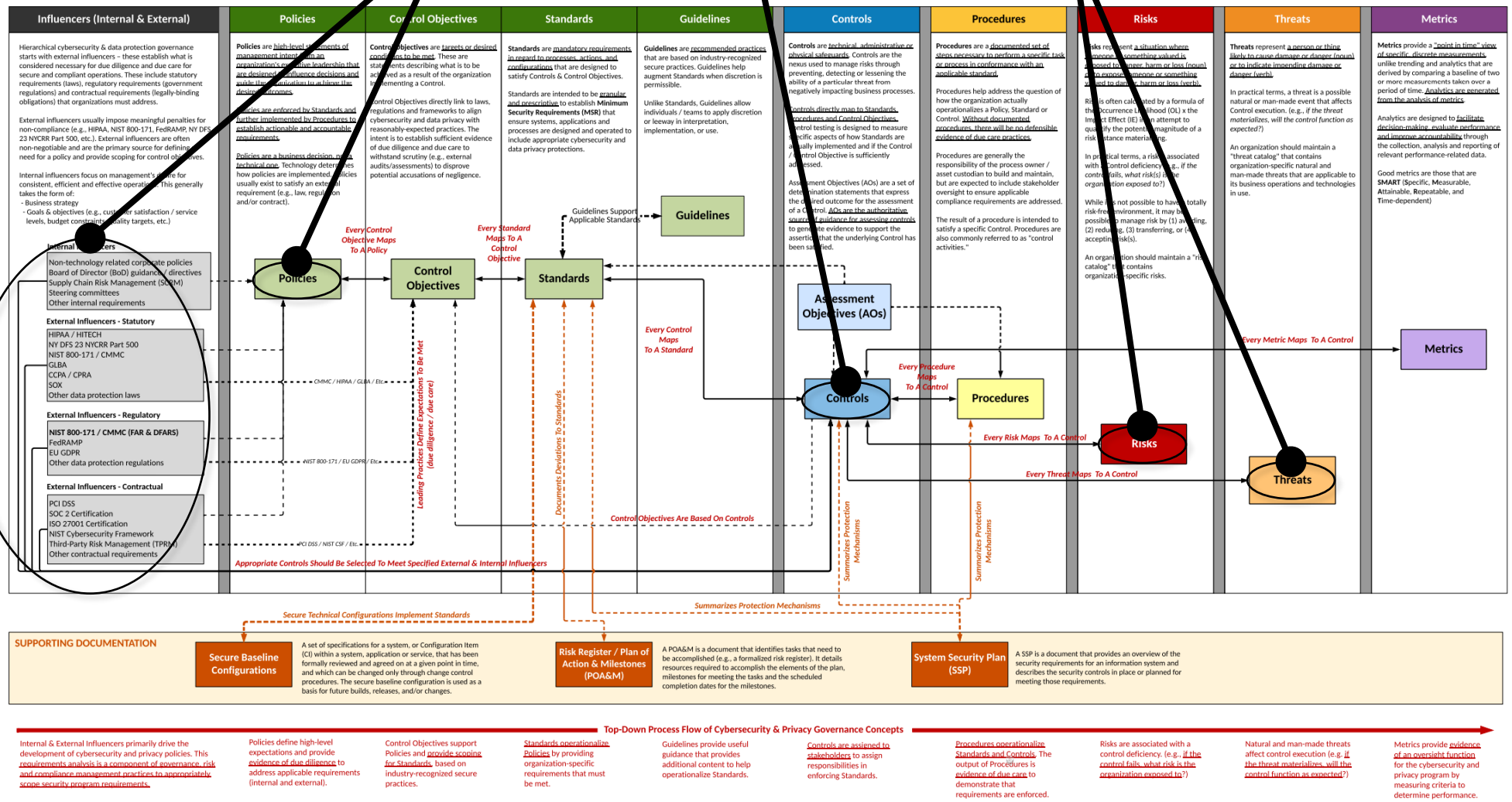


How do we actually succeed then?

ALWAYS LEADING

COMPLIANCE FORGE

Operations, Security, & Risk all need to align



SENTINEL

Linkedin.com/in/mikesoule/

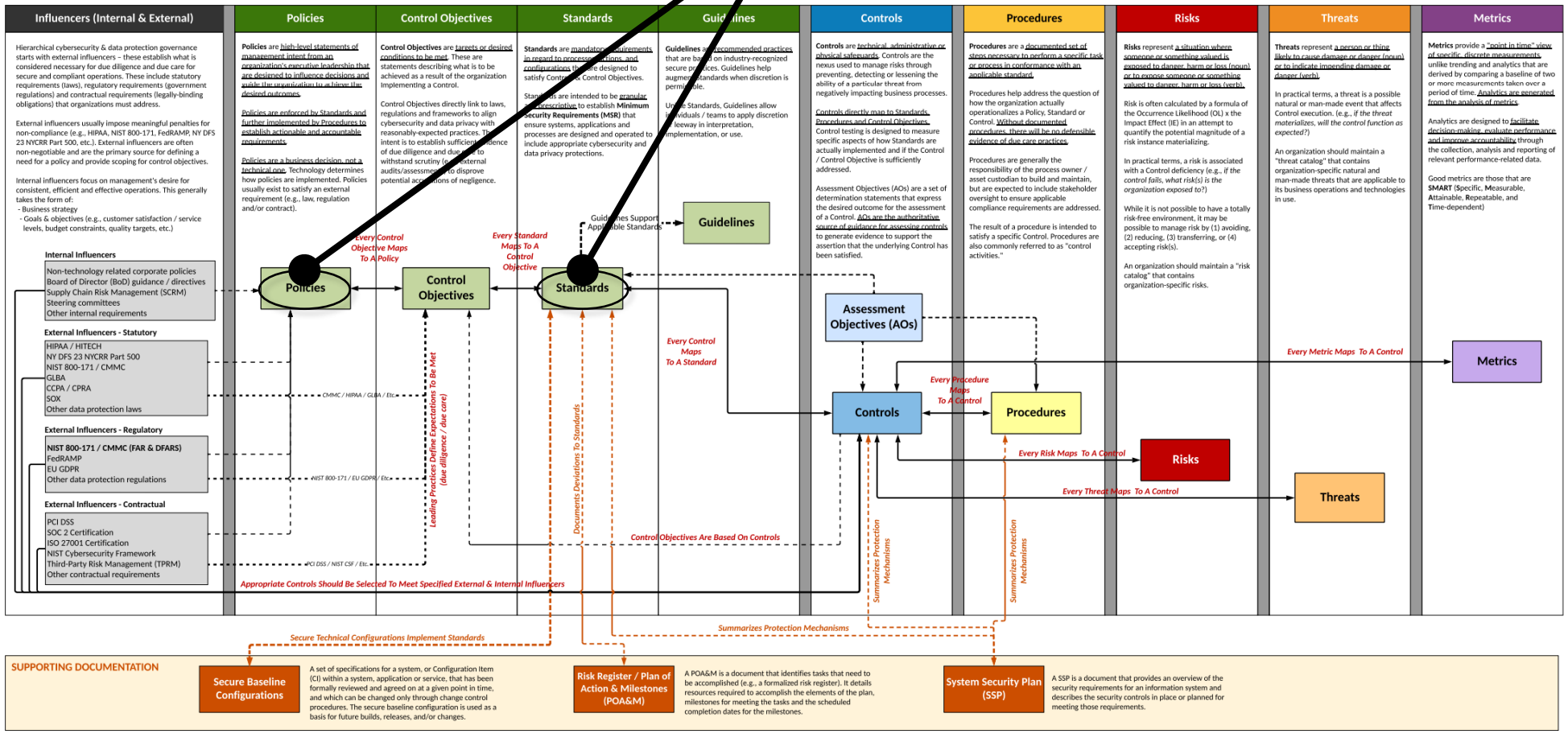
Research Triangle Park PowerShell User Group

Quick opinion, your organization, likely, doesn't author standards…

ALWAYS LEADING

COMPLIANCE FORGE

Technical Standards VS Risk Management Standards



SENTINEL

Linkedin.com/in/mikesoule/

Research Triangle Park PowerShell User Group

MITRE ATT&CK®

- **Command and Scripting Interpreter: PowerShell** (T1059.001)
  Adversaries may abuse PowerShell commands and scripts for execution.
  - Antivirus/Antimalware (M1049)
  - Code Signing (M1045)
  - Disable or Remove Feature or Program (M1042)
  - Execution Prevention (M1038)
  - Privileged Account Management (M1026)
  - Abuse of PowerShell for Arbitrary Execution (DET0455)

- **Command and Scripting Interpreter: PowerShell** (T1059.001)
  Adversaries may abuse PowerShell commands and scripts for execution.
  - Antivirus/Antimalware (M1049)
  - Code Signing (M1045)
  - **Disable or Remove Feature or Program** (M1042)
  - Execution Prevention (M1038)
  - Privileged Account Management (M1026)
  - Abuse of PowerShell for Arbitrary Execution (DET0455)

MITRE | ATT&CK®

- **Command and Scripting Interpreter: PowerShell** (T1059.001)
  Adversaries may abuse PowerShell commands and scripts for execution.
  - Antivirus/Antimalware (M1049)
  - Code Signing (M1045)
  - Disable or Remove Feature or Program (M1042)
  - Execution Prevention (M1038)
  - Privileged Account Management (M1026)
  - Abuse of PowerShell for Arbitrary Execution (DET0455)
- **Event Triggered Execution: PowerShell Profile** (T1546.013)
  Adversaries may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles.
  - Code Signing (M1045)
  - Restrict File and Directory Permissions (M1022)
  - Software Configuration (M1054)
  - Detection Strategy for PowerShell Profile Persistence via profile.ps1 Modification (DET0451)

How do I limit PowerShell risks?

- Anything Could Have Happened
- Anyone Can Run PowerShell
- Any Code Can Run
- Access from Anywhere to Anywhere
- Developers Make Mistakes

How do I limit PowerShell risks?

- Anything Could Have Happened
  - Audit Logging
  - Endpoint Detection & Response (EDR)
- Anyone Can Run PowerShell

- Any Code Can Run

- Access from Anywhere to Anywhere

- Developers Make Mistakes

How do I limit PowerShell risks?

- Anything Could Have Happened
  - Audit Logging
  - Endpoint Detection & Response (EDR)
- Anyone Can Run PowerShell
  - Administrative Tiers (ESAE, RaMP)
    - Including Non-Human Identities
  - Software Restrictions (WDAC, AppLocker, SRP)
- Any Code Can Run

- Access from Anywhere to Anywhere

- Developers Make Mistakes

How do I limit PowerShell risks?

- Anything Could Have Happened
  - Audit Logging
  - Endpoint Detection & Response (EDR)
- Anyone Can Run PowerShell
  - Administrative Tiers (ESAE, RaMP)
    - Including Non-Human Identities
  - Software Restrictions (WDAC, AppLocker, SRP)
- Any Code Can Run
  - Script Signing (Execution Policy)
  - Delegation Management (JEA)
- Access from Anywhere to Anywhere

- Developers Make Mistakes

How do I limit PowerShell risks?

- Anything Could Have Happened
  - Audit Logging
  - Endpoint Detection & Response (EDR)
- Anyone Can Run PowerShell
  - Administrative Tiers (ESAE, RaMP)
    - Including Non-Human Identities
  - Software Restrictions (WDAC, AppLocker, SRP)
- Any Code Can Run
  - Script Signing (Execution Policy)
  - Delegation Management (JEA)
- Access from Anywhere to Anywhere
  - Inbound (WinRM, SSH)
    - Including Authentication Challenges
  - Outbound (Host Firewall)
- Developers Make Mistakes

How do I limit PowerShell risks?

- Anything Could Have Happened
  - Audit Logging
  - Endpoint Detection & Response (EDR)
- Anyone Can Run PowerShell
  - Administrative Tiers (ESAE, RaMP)
    - Including Non-Human Identities
  - Software Restrictions (WDAC, AppLocker, SRP)
- Any Code Can Run
  - Script Signing (Execution Policy)
  - Delegation Management (JEA)
- Access from Anywhere to Anywhere
  - Inbound (WinRM, SSH)
    - Including Authentication Challenges
  - Outbound (Host Firewall)
- Developers Make Mistakes
  - Secret Management, AppSec Testing

Topic (MITRE ID)	Effort (High/Low)	Benefit (High/Low)	
Audit Logging (M1054)	Low	High	Existing Resources and Processes
EDR (M1049)	Low	High	
Remoting (M1054)	Low	High	
AppSec*	Low	Low	Requires New Resources or Processes
Script Signing* (M1045)	High	High	
Admin Tiering (M1026)	High	High	
Host Firewall	High	High	
Secret Management*	High	High	
Software Restrictions* (M1042)	High	High	
Delegation Management (M1038)	High	Low	

* All components of Software Development Life Cycle (SDLC)

https://www.cyber.gov.au/business-government/protecting-devices-systems/system-administration/securing-powershell-in-the-enterprise

Michael Soule
National Director
Sentinel Technologies
misoule@sentinel.com

SENTINEL®

FORTIS By Sentinel
Advancing Your Protection

- Migration & Modernization
- Identity & Security
- Hybrid Cloud
- Licensing & Cost Optimization

Topic (MITRE ID)	Effort (High/Low)	Benefit (High/Low)	
Audit Logging (M1054)	Low	High	Existing Resources and Processes
EDR (M1049)	Low	High	
Remoting (M1054)	Low	High	
AppSec*	Low	Low	Requires New Resources or Processes
Script Signing* (M1045)	High	High	
Admin Tiering (M1026)	High	High	
Host Firewall	High	High	
Secret Management*	High	High	
Software Restrictions* (M1042)	High	High	
Delegation Management (M1038)	High	Low	

* All components of Software Development Life Cycle (SDLC)

SENTINEL®

NEVER FOLLOW

At Sentinel, we've always taken the lead. Since 1982, Sentinel Technologies has been recognized as the premier business technology services provider dedicated to delivering the highest IT solutions, customer service and support. As technology evolves, Sentinel remains at the forefront of IT developments and maintains its singular focus of providing practical and innovative solutions.

When it comes to achieving a tangible ROI and sustainable performance from your IT environment. Sentinel takes the lead!

THE SENTINEL DIFFERENCE

ALWAYS LEADING

Deep breadth of technical expertise

Commitment to thought leadership

50+ company certifications & Specializations

2,400+ individual certifications

Continuous skills advancement

World-class customer satisfaction

Award winning service

24*7 maintenance, monitoring & managed support

ALWAYS MOTIVATED

Awarded “Chicago’s Best 100 Places to Work” by The Chicago Tribune

Crain’s “Best Places to Work”

Arizona top workplace every year since 2015

Great Place to Work Certified

SENTINEL

ALWAYS ENGAGED

Detailed schedules ensure project timing

EPMO utilizing advanced ServiceNow project tracking

Waterfall and agile approaches

Follow OPM3 standards

Award-winning service delivery

PMP certified project management

SENTINEL®

SENTINEL CORE SOLUTIONS

Sentinel understands your business is in constant motion. We take the initiative to make solutions happen that drive your business forward.

- Consulting & Advisory Services
- Collaboration and Modern Workplace
- Public Cloud, Hybrid Cloud & Data Center
- Cyber Security Solutions and
- 24*7 Maintenance and Managed Services
- Managed Detection & Response with 24*7 Security Operations (SOC)
- Always Ready Incident Response

Vertical Specialties

- Healthcare Innovations