Windows Server 2016 Security

Igor Shastitko, IT infrastructure Consultant

Microsoft

Microsoft

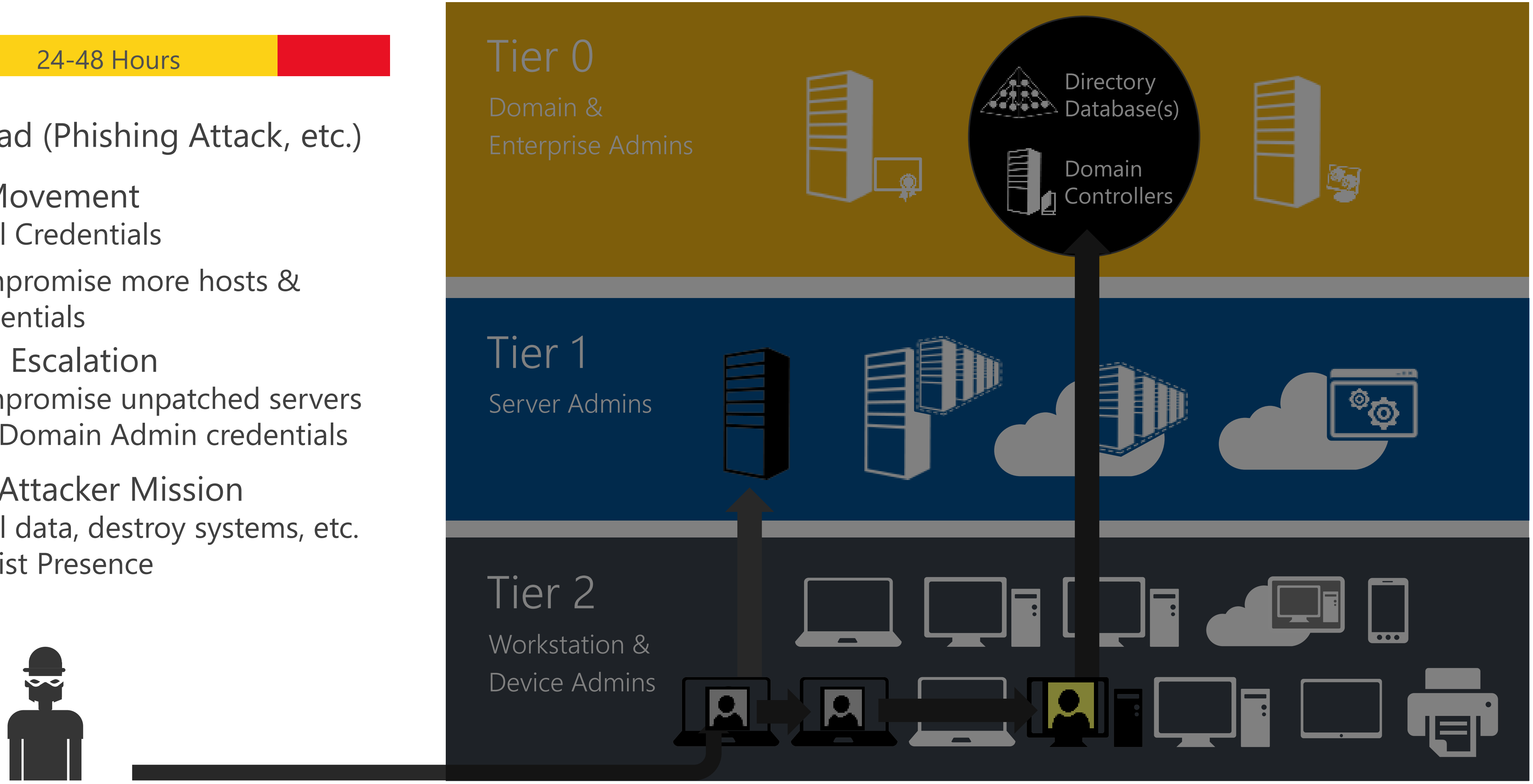
Securing Privileged Access

A practical roadmap

Typical Credential Theft Attack

Compromises administrative control

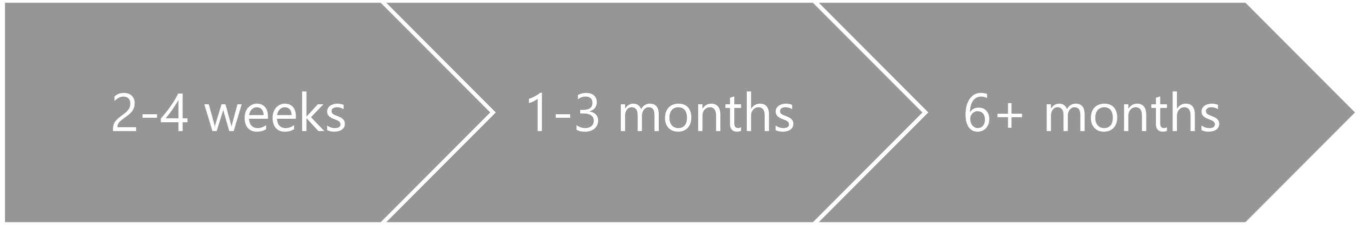
1. Beachhead (Phishing Attack, etc.)
2. Lateral Movement
   a. Steal Credentials
   b. Compromise more hosts & credentials
3. Privilege Escalation
   a. Compromise unpatched servers
   b. Get Domain Admin credentials
4. Execute Attacker Mission
   a. Steal data, destroy systems, etc.
   b. Persist Presence



How to protect your privileges against these attacks

Attack	Defense
Credential Theft & Abuse	Prevent Escalation
	Prevent Lateral Traversal
	Increase Privilege Usage Visibility
DC Host Attacks	Harden DC configuration
	Reduce DC Agent attack surface
AD Attacks	Assign Least Privilege
Attacker Stealth	Detect Attacks

Three Stage Mitigation Plan



2-4 weeks

1-3 months

6+ months

http://aka.ms/privsec

These practices are still important

Part of a complete long term security strategy

Domain Controller Security Updates

Target full deployment within 7 days

Remove Users from Local Administrators

Manage exceptions down to near-zero
Ensure only admin of one workstation

Baseline Security Policies

Apply standard configurations
Manage exceptions down to near-zero

Anti-Malware

Detect and clean known threats

Log Auditing and Analysis

Centralize logs to enable investigations and analysis

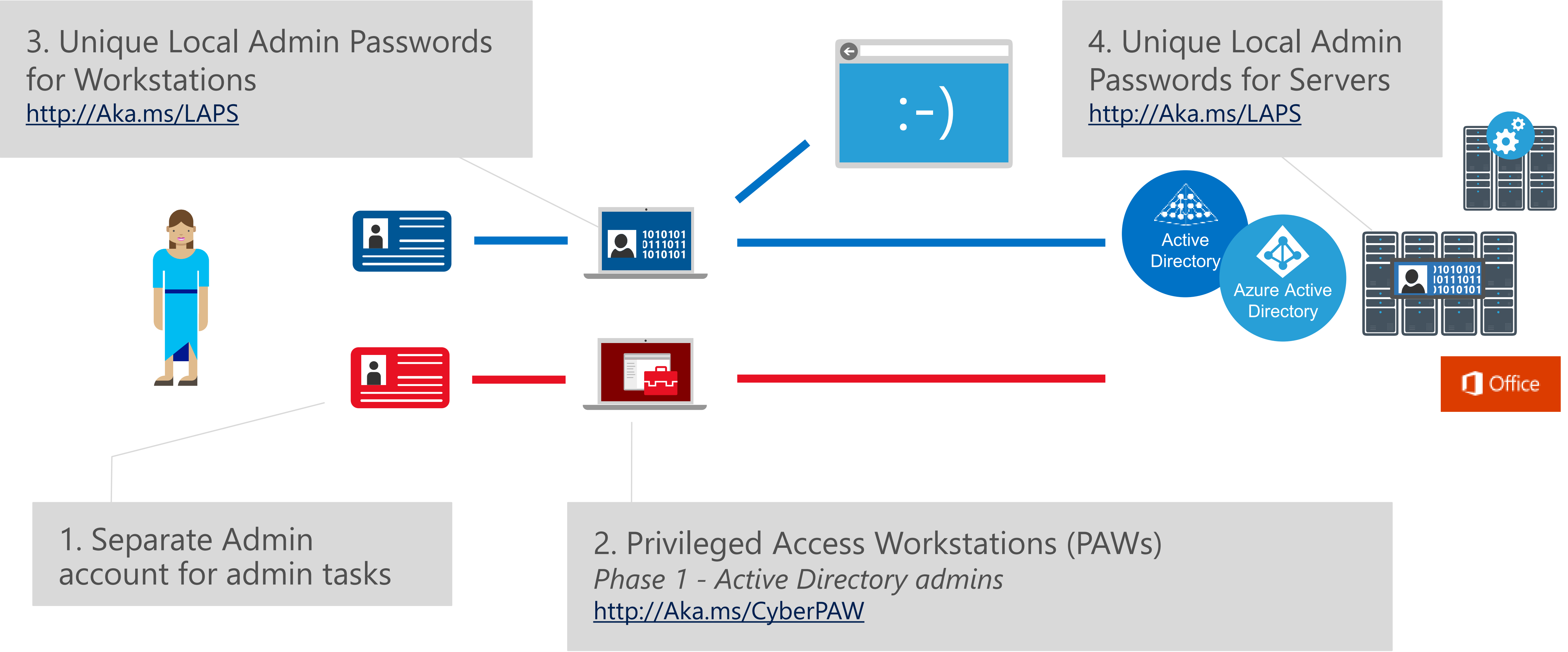
Software Inventory and Deployment

Ensure visibility and control of endpoints to enable security operations

Protecting Active Directory and Admin privileges

2-4 weeks → 1-3 months → 6+ months

First response to the most frequently used attack techniques



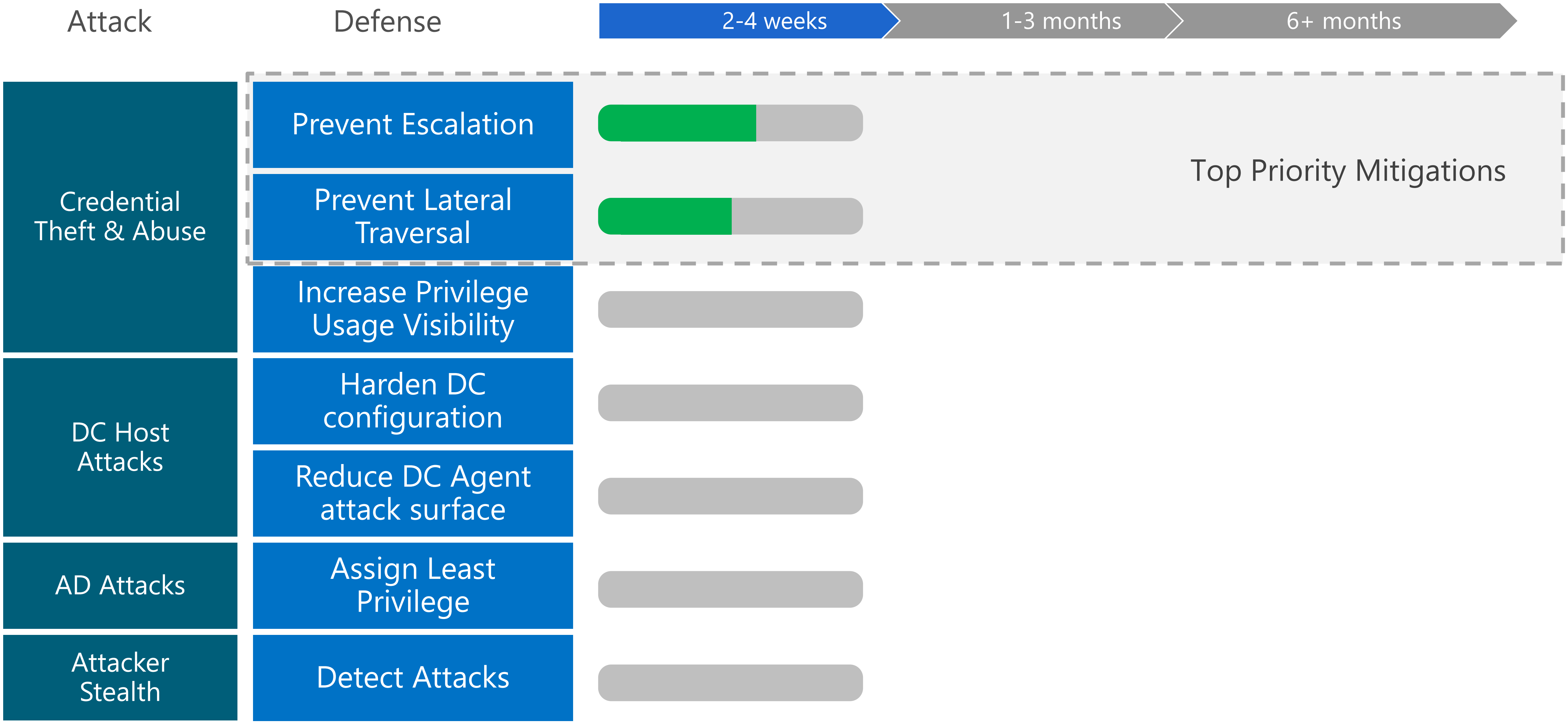
3. Unique Local Admin Passwords for Workstations
http://Aka.ms/LAPS

4. Unique Local Admin Passwords for Servers
http://Aka.ms/LAPS

1. Separate Admin account for admin tasks

2. Privileged Access Workstations (PAWs)
Phase 1 - Active Directory admins
http://Aka.ms/CyberPAW

First response to the most frequently used attack techniques



2-4 weeks

1-3 months

6+ months

Attack	Defense
Credential Theft & Abuse	Prevent Escalation
	Prevent Lateral Traversal
	Increase Privilege Usage Visibility
DC Host Attacks	Harden DC configuration
	Reduce DC Agent attack surface
AD Attacks	Assign Least Privilege
Attacker Stealth	Detect Attacks

Top Priority Mitigations

Protecting Active Directory and Admin privileges

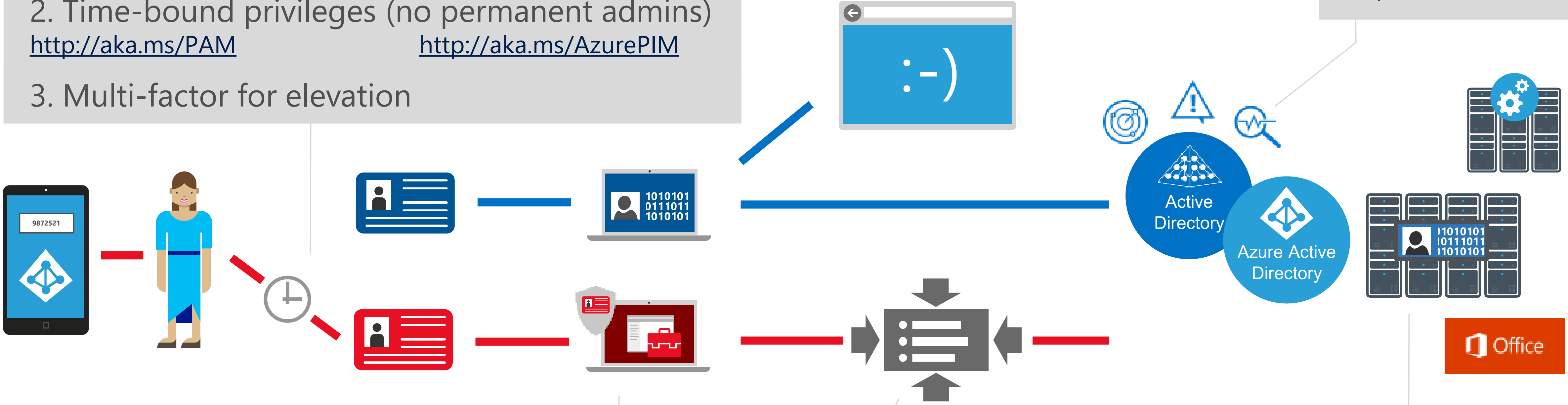
2-4 weeks › **1-3 months** › 6+ months

Build visibility and control of administrator activity, increase protection against typical follow-up attacks

2. Time-bound privileges (no permanent admins)
http://aka.ms/PAM http://aka.ms/AzurePIM

3. Multi-factor for elevation

6. Attack Detection
http://aka.ms/ata



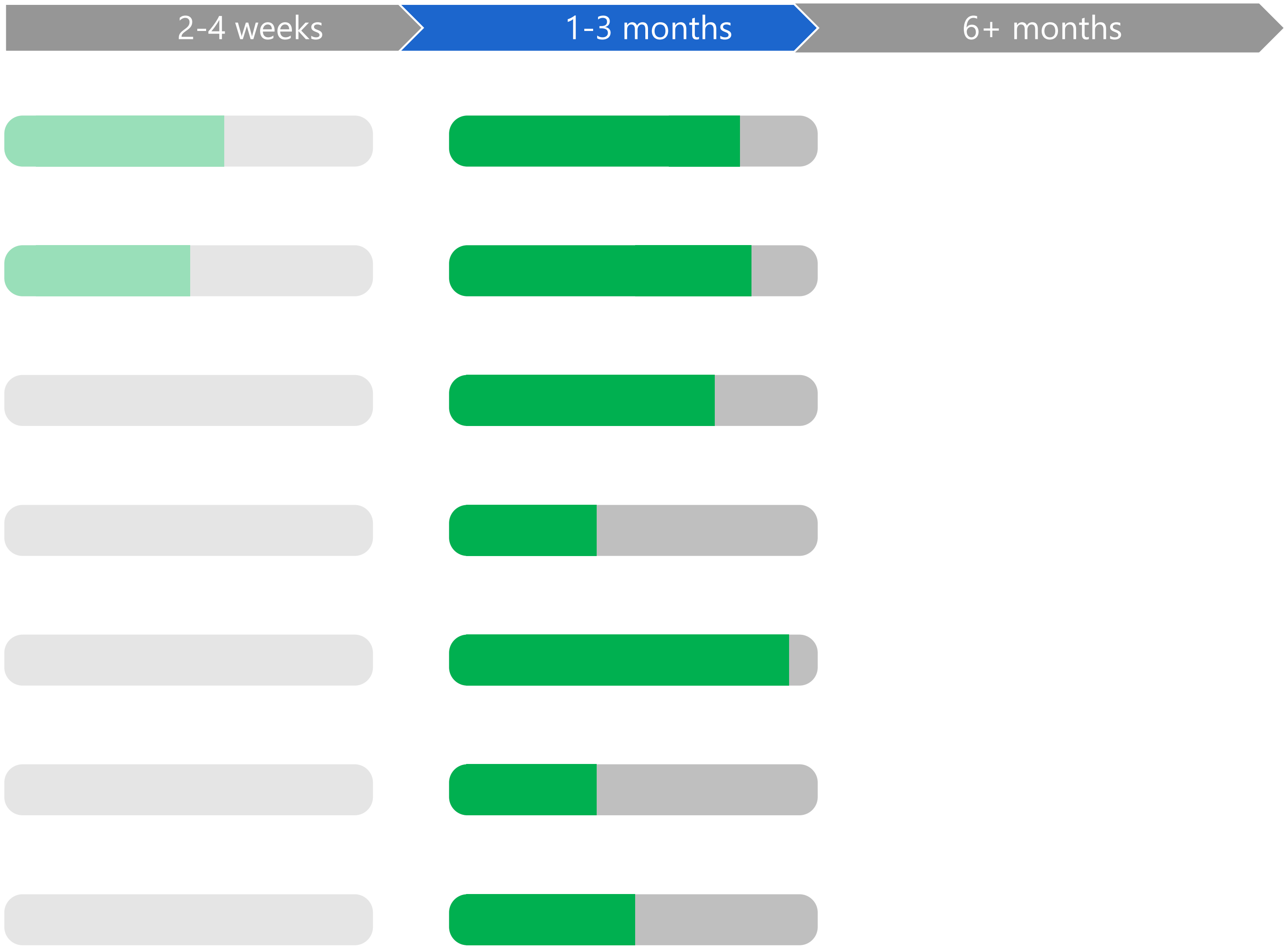
1. Privileged Access Workstations (PAWs)
Phases 2 and 3 –All Admins and additional hardening (Credential Guard, RDP Restricted Admin, etc.)
http://aka.ms/CyberPAW

4. Just Enough Admin (JEA) for DC Maintenance
http://aka.ms/JEA

5. Lower attack surface of Domain and DCs
http://aka.ms/HardenAD

Build visibility and control of admin activity

Attack	Defense
Credential Theft & Abuse	Prevent Escalation
	Prevent Lateral Traversal
	Increase Privilege Usage Visibility
DC Host Attacks	Harden DC configuration
	Reduce DC Agent attack surface
AD Attacks	Assign Least Privilege
Attacker Stealth	Detect Attacks

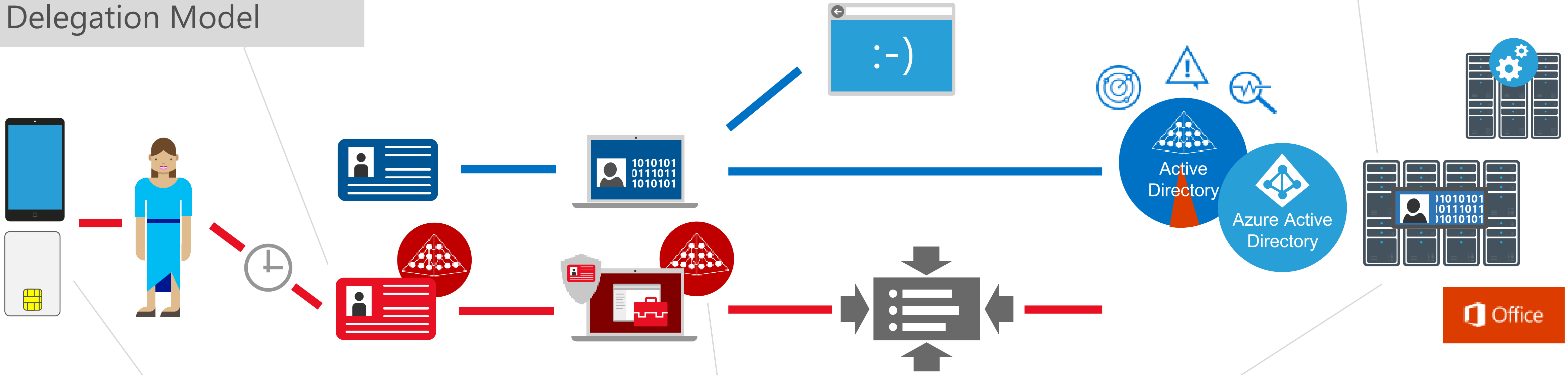


Protecting Active Directory and Admin privileges

2-4 weeks > 1-3 months > 6+ months

Move to proactive security posture

1. Modernize Roles and Delegation Model



2. Smartcard or Passport Authentication for all admins
http://aka.ms/Passport

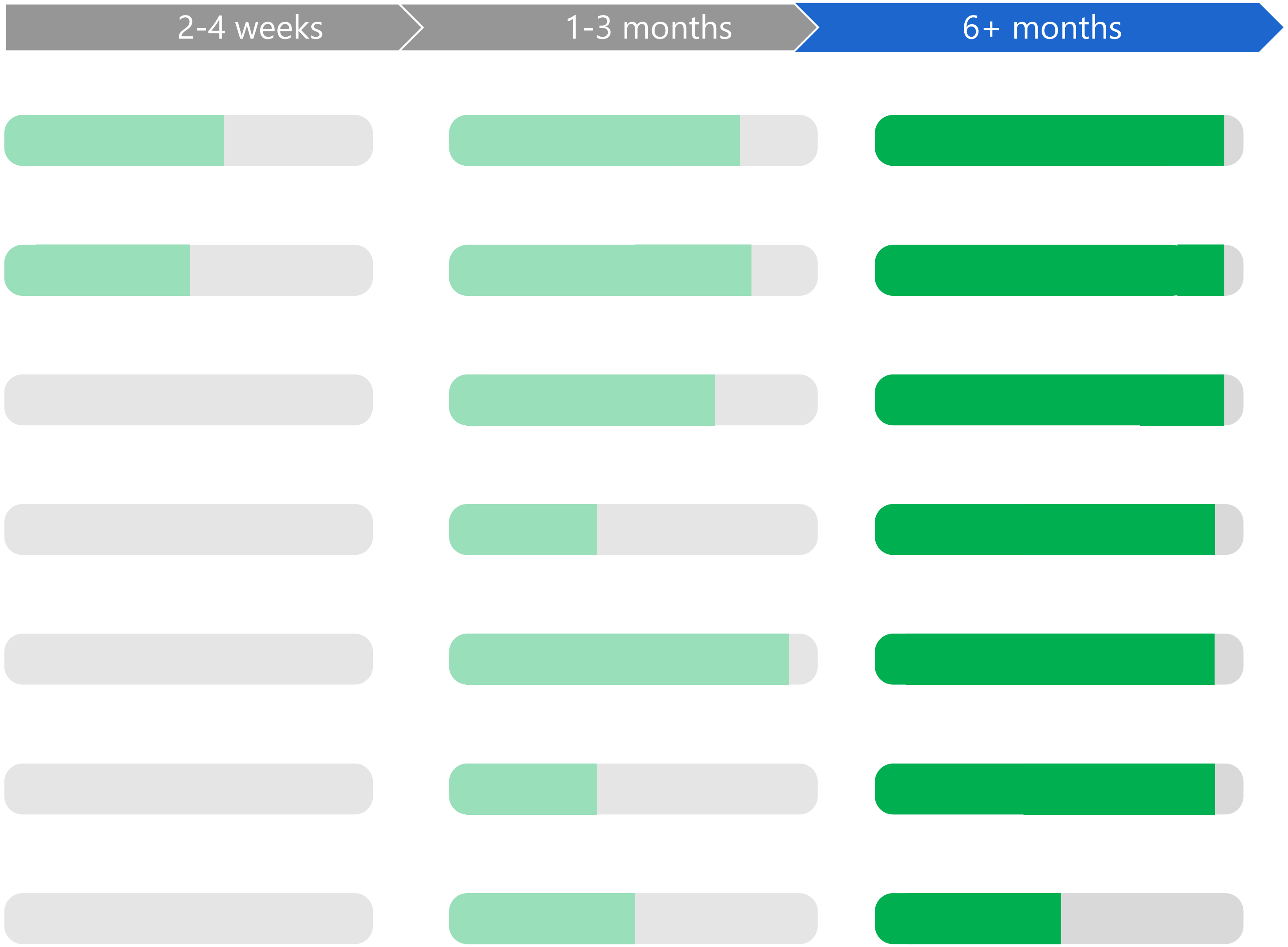
3. Admin Forest for Active Directory administrators
http://aka.ms/ESAE

4. Code Integrity Policy for DCs (Server 2016)

5. Shielded VMs for virtual DCs (Server 2016 Hyper-V Fabric)
http://aka.ms/shieldedvms

Move to proactive security posture

Attack	Defense
Credential Theft & Abuse	Prevent Escalation
	Prevent Lateral Traversal
	Increase Privilege Usage Visibility
DC Host Attacks	Harden DC configuration
	Reduce DC Agent attack surface
AD Attacks	Assign Least Privilege
Attacker Stealth	Detect Attacks



Microsoft

JEA: Just Enough Admin

JEA: Just Enough Admin

- Based on the PowerShell security features used by online services
  - Enabled remote administration of Exchange Online
- Simple concepts
  - Role Capabilities
    - Refined set of commands to support the activities of a specific user role
  - Endpoint
    - Management connection point where authorized users are provided the appropriate role capabilities
  - Identity
    - Privileged alternate identity used to invoke commands

Role Capabilities

```
@{

# Description of the functionality provided by these settings
Description = 'Role Capabilities for DNS Maintenance'

# Modules to import when applied to a session
ModulesToImport = 'DnsServer'

# Cmdlets to make visible when applied to a session
VisibleCmdlets = 'Get-Service', 'Restart-Service',
'Get-DnsServerCache', 'Clear-DnsServerCache',
'Show-DnsServerCache'

# Functions to define when applied to a session
FunctionDefinitions = @{
    'Name' = 'whoami'
    'ScriptBlock' = { $PSSenderInfo } }

}
```

Session Configuration

```
@{

# Session type defaults to apply for this session configuration.
# Can be 'RestrictedRemoteServer' (recommended), 'Empty', or 'Default'
SessionType = 'RestrictedRemoteServer'

# Directory to place session transcripts for this session configuration
TranscriptDirectory = 'C:\Program Files\Endpoints\DnsMaintenance\Transcripts'

# Whether to run this session configuration as the machine's
# (virtual) administrator account
RunAsVirtualAccount = $true

# User roles (security groups), and the role capabilities
# that should be applied to them when applied to a session
RoleDefinitions = @{
    'DnsAdmin' = @{
        'RoleCapabilities' = 'DnsMaintenance' } }

}
```

Identity

- Who's actually running the commands in a JEA session?

Identity Type	Description
Connected User (Default)	Hosting process runs under the connected user's identity.
Named Identity	Hosting process runs under the credentials of a specific account.
Virtual Account	Hosting process runs under a local temporary administrative identity.
Group Managed Service Account (GMSA)*	Hosting process runs under a managed domain identity that has its password automatically managed and rotated by Active Directory.

Why PowerShell?

- JEA is about controlling admin actions
- Like all shells, PowerShell dispatches commands
  - You can control what gets dispatched by traditional things like path, loading policy, etc.
  - PowerShell adds **command visibility**
- Unlike many shells, PowerShell also does command parsing!
- Parsing is driven off of data structures
  - Which you can program
  - Which you can program to create **proxies**
- Command visibility and proxies allow us to secure the environment

Creating a Proxy Command

PowerShell owns the Parser

```
$cmd        = Get-Command Stop-Process
$MetaData  = New-Object System.Management.Automation.CommandMetaData $cmd
```

You can program a cmdlet’s parameters

```
$MetaData.Parameters.Remove("ID")
$MetaData.Parameters.Name.Attributes.Add((New-Object `
 System.Management.Automation.ValidateSetAttribute ("notepad","calc")))
$MetaData.DefaultParameterSetName ="Name"
```

And then publish a proxy

```
${Function:Stop-Process} =
[System.Management.Automation.ProxyCommand]::create($MetaData)
```

Now hide the original

```
$cmd.Visibility = "private"
```

Fine-Grained Proxy Control

```
# Cmdlets to make visible when applied to a session
VisibleCmdlets = 'Get-Service', 'Get-DnsServerCache',
    'Clear-DnsServerCache', 'Show-DnsServerCache',
    @{
        Name = 'Restart-Service'
        Parameters = @{
            Name = 'Name'
            ValidateSet = 'DNS','DNSCache'
        }
    }
```

Creating a Constrained PowerShell Configuration

- New-PSRoleCapabilityFile –Path DnsAdmins.psrc -<…>
- New-PSSessionConfigurationFile –Path DnsMaintenance.pssc -<…>

- Register-PSSessionConfiguration -Path DnsMaintenance.pssc
- Enter-PSSession -ComputerName Server1

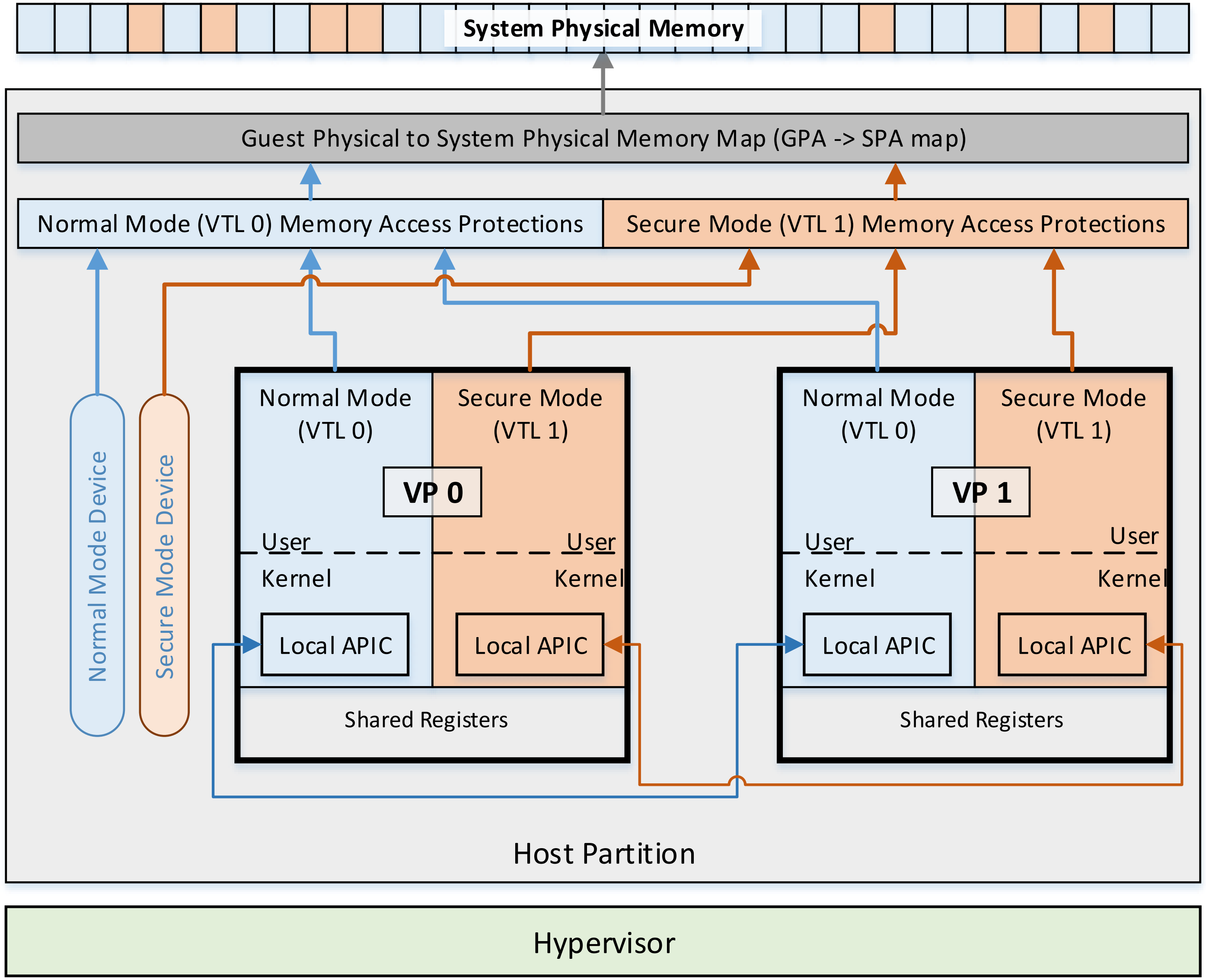
Microsoft

Virtual Secure Mode

Virtual Secure Mode (VSM)

- VSM is Microsoft’s virtualization-based security solution
- VSM provides the basis for:
  - Secure runtime environment
  - Protected store
- VSM is available to both server and client systems
- No user interaction with VSM

VSM protections



Virtual Secure Mode (VSM)

- Provides a new trust boundary for system software to:
  - Enhance platform security
  - Leverage platform virtualization to enforce strong access guarantees
  - Limit access to high-value security assets, even from code running in kernel mode
- Provide a secure store and execution environment to enable:
  - Protected storage and management of platform security assets
  - Enhanced OS protection against attacks (including attacks from kernel-mode)
  - A basis for strengthening protections of guest VM secrets from the host OS

Uses for Virtual Secure Mode

- Protect security assets
  - Authenticated user credentials
  - Security keys
  - Security policy
  - Attestation logs
- Host services in isolation from the normal OS environment
  - Windows Security Applications (WSA) or “trustlets”
  - Code integrity enforcement
  - Attestation of host health

VSM platform recommendations

- Virtualization extensions (Intel VT-x)
- Second Level Address Translation (Intel Extended Page Tables, aka EPT)
- IOMMU (Intel VT-d)
- UEFI 2.3.1c or higher
- Secure Boot
- TPM v2.0

VSM memory isolation

- VSM provides memory isolation from:
  - Accesses generated by system processors
  - DMA initiated from devices
- Memory isolation is:
  - Based on Virtual Trust Levels, each with its own set of address space protections
  - Enforced by the hypervisor

Virtual Trust Levels

- VSM implements trust boundaries via Virtual Trust Levels (VTL)
  - VTLs enhance existing processor privilege levels
- VTLs provide memory isolation
  - Essentially, a set of access protections on physical memory
  - Enforced during the partition’s physical memory translation
- VTLs cannot be changed from CPL0 in the partition

Virtual Trust Levels

- VTLs are hierarchical
  - Higher trust level == greater privilege level
- Two trust levels for the initial VSM implementation:
  - VTL0 – Normal Mode, VTL1 – Secure Mode
  - Design accommodates > 2 VTLs
- Higher VTLs control access privileges for lower levels
  - VTL1 can adjust memory access protections for VTL0

VSM memory isolation

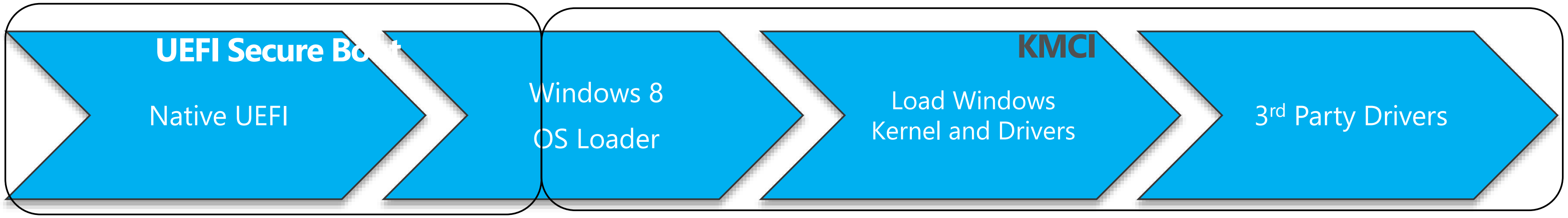
- Isolation from device accesses (DMA) enforced via IOMMU
- Normal Mode devices share Normal Mode memory access protections
    - Cannot access Secure Mode memory

Microsoft

Hypervisor Enforced Code Integrity

Code Integrity Enforcement

- Secure Boot
  - Ensures that everything that boots on a platform is signed by a trusted authority
  - Includes Secure Firmware Updates and "Platform" Secure Boot
- Kernel Mode Code Integrity (KMCI)
  - Feature in Windows that ensures that any code running in kernel is signed by a trusted authority



Hypervisor Enforced Code Integrity

- Currently CI enforcement is done from within the Kernel
- If the Kernel is compromised any code can be executed
- For Hypervisor CI (HVCI) based systems enforcement will be in VSM
- Pages can only be marked executable from VSM after verification in VSM
- Eliminates most memory based attacks
- Reduces risk from 3’rd party drivers

How can I protect my datacenter assets from emerging threats?

Protect your virtual infrastructure from emerging threats

Hardware-rooted security for zero-trust environments

Host Guardian Service

Guarded Hosts

Shielded VM

Virtual secure mode

Need to maintain stewardship of corporate assets in the midst of emerging threats

1 Increasing incidents

2 Bigger motivations

3 Bigger risk

Cyberattacks on the rise against US corporations

New York Times [2014]

1

Espionage malware infects rafts of governments, industries around the world

Ars Technica [2014]

1

Cybercrime costs US economy up to $140 billion annually, report says

Los Angeles Times [2014]

2

How hackers allegedly stole “unlimited” amounts of cash from banks in just a few hours

Ars Technica [2014]

2

The biggest cyberthreat to companies could come from the inside

Cnet [2015]

3

Malware burrows deep into computer BIOS to escape AV

The Register [September 2014]

3

Forget carjacking, soon it will be carhacking

The Sydney Morning Herald [2014]

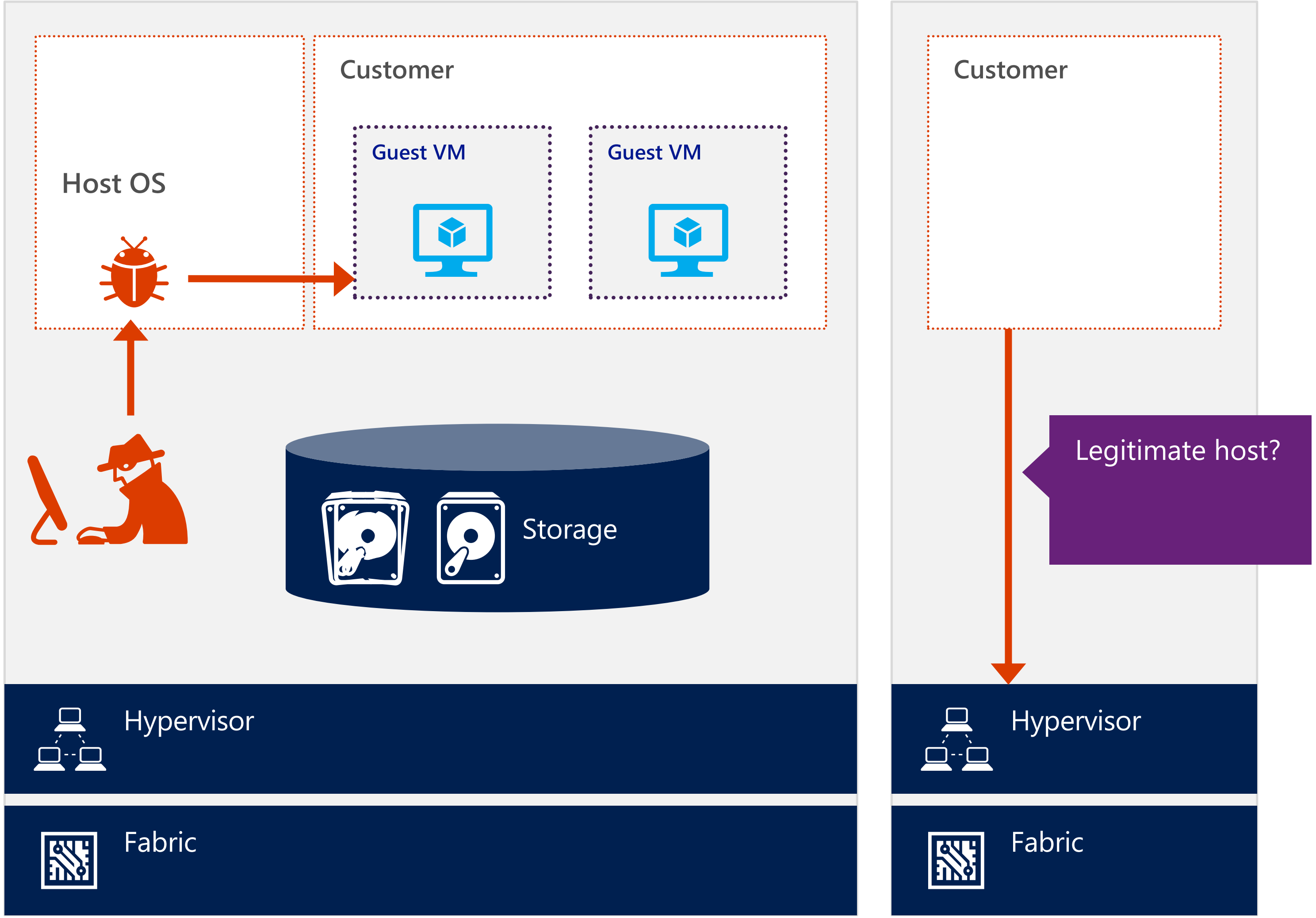
3

Challenges in protecting high-value assets

Any seized or infected host administrators can access guest virtual machines

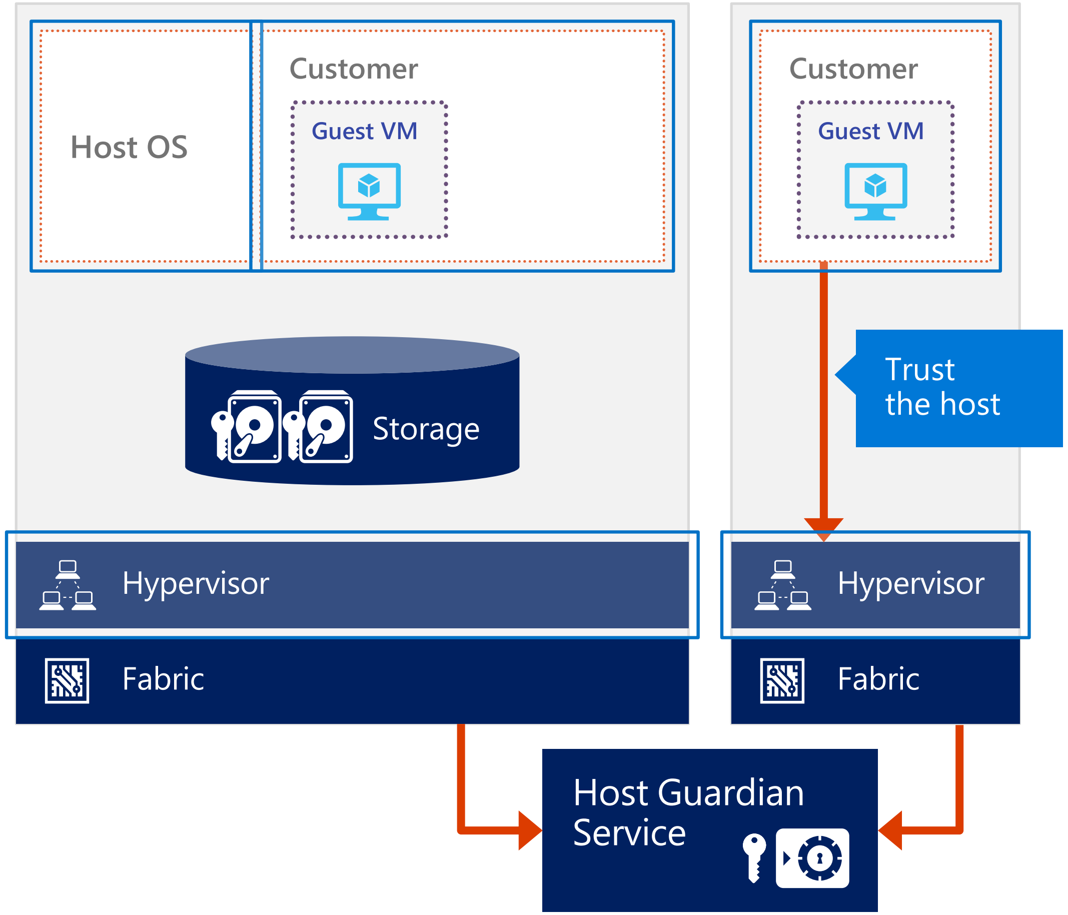
Impossible to identify legitimate hosts without a hardware based verification

Tenants VMs are exposed to storage and network attacks while unencrypted



Confidently protect sensitive customer data: Designed for 'zero-trust' environments

Hardware-rooted technologies to separate the guest operating system from host administrators	**Virtual Secure Mode** Process and Memory access protection from the host
Guarded fabric to identify legitimate hosts and certify them to run shielded tenant Generation 2 VMs	**Host Guardian Service** Enabler to run Shielded Virtual Machines on a legitimate host in the fabric
Virtualized trusted platform module (vTPM) support to encrypt virtual machines	**Shielded VM** Bitlocker enabled VM



Windows Server Hyper-V Microsoft System Center

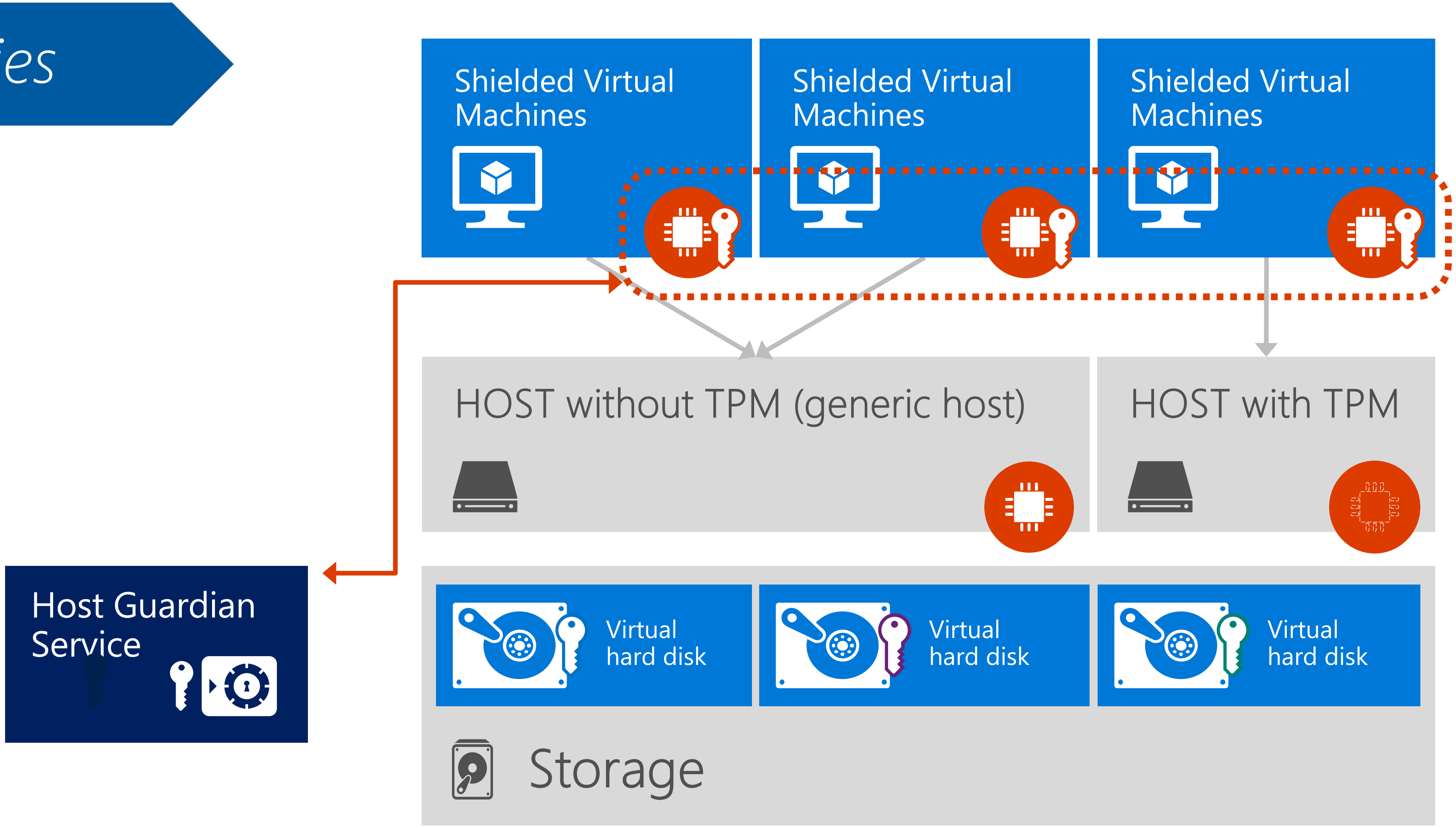
Shielded VMs

Spotlight capabilities

Shielded Virtual Machines can only run in fabrics that are designated as owners of that virtual machine

Shielded Virtual Machines will need to be **encrypted** (by **BitLocker** or other means) in order to ensure that only the designated owners can run this virtual machine

You can **convert** a **running Generation 2 virtual machine** into a Shielded Virtual Machine


Shielded Virtual Machines
Shielded Virtual Machines
Shielded Virtual Machines
HOST without TPM (generic host)
HOST with TPM
Host Guardian Service
Virtual hard disk
Virtual hard disk
Virtual hard disk
Storage

Hyper-V Microsoft System Center Windows Server

Nano Server

Minimum-footprint infrastructure OS and application OS

'Cloud-first' refactoring

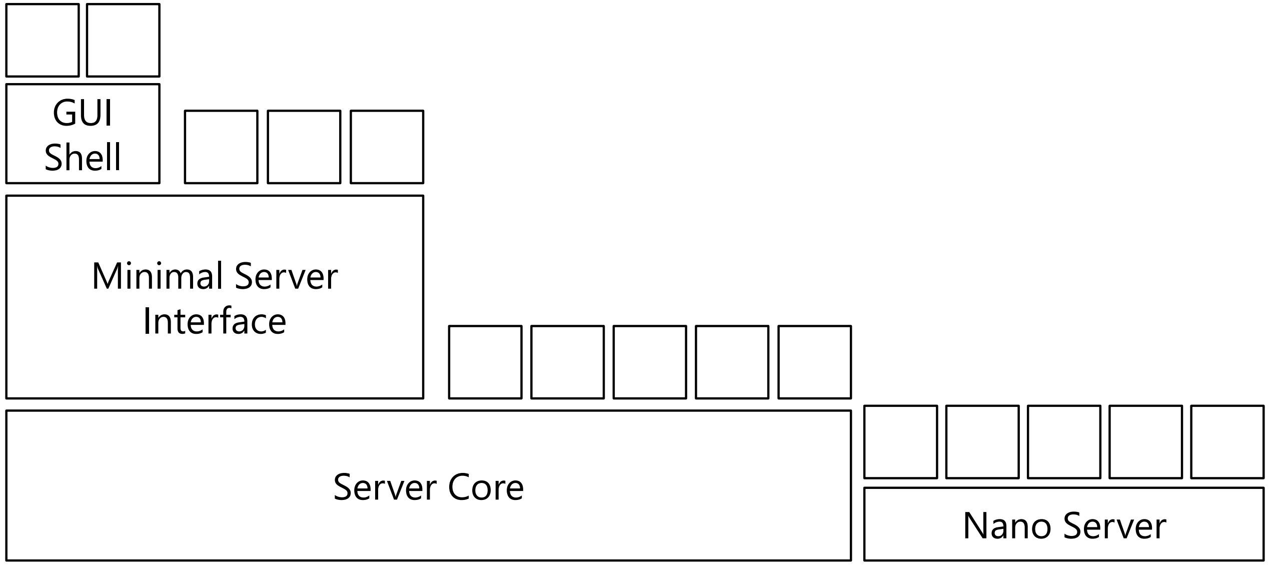
Infrastructure:

Hyper-V, Storage, Clustering

Application:

Next-gen application platform and run-time

Containers



Windows Server 2016

Nano Server – Just enough OS

Nucleus of next-gen cloud infrastructure and applications

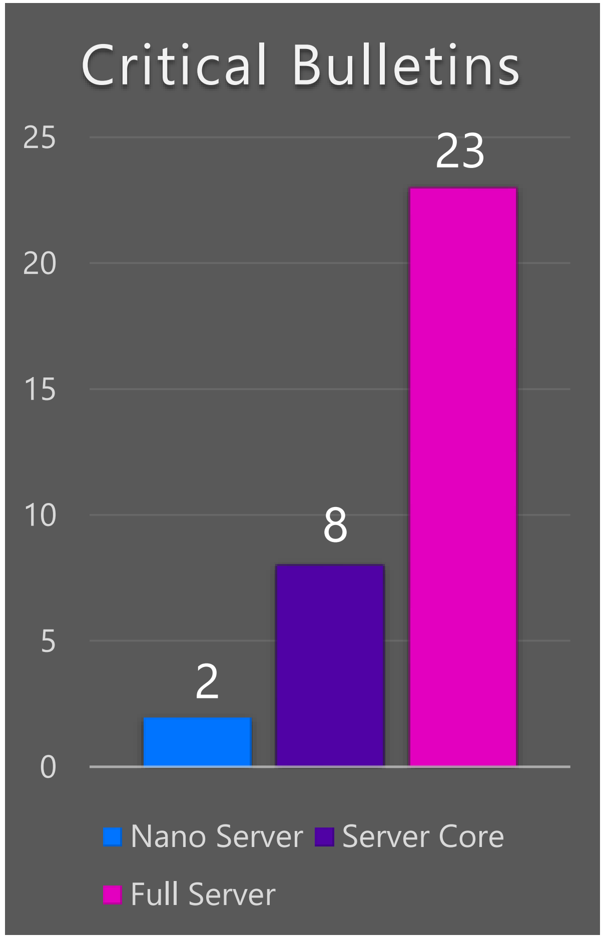
Powers modern cloud infrastructure

- Faster time to value
- Much lower servicing footprint
- Significantly lower attack surface
- Breakthrough efficiency

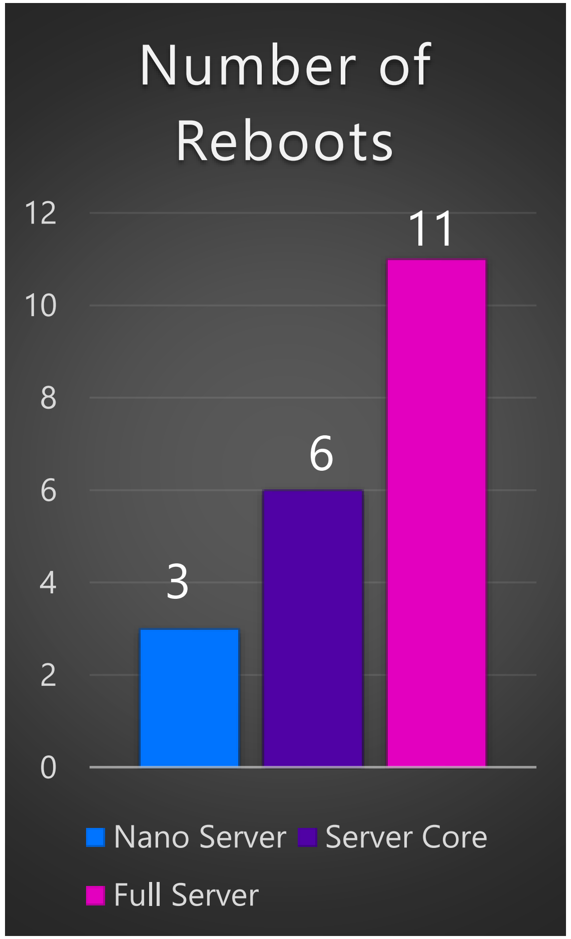
Optimized for next-gen distributed applications

- Higher density and performance
- Next-gen distributed app frameworks
- Interoperate with existing server applications

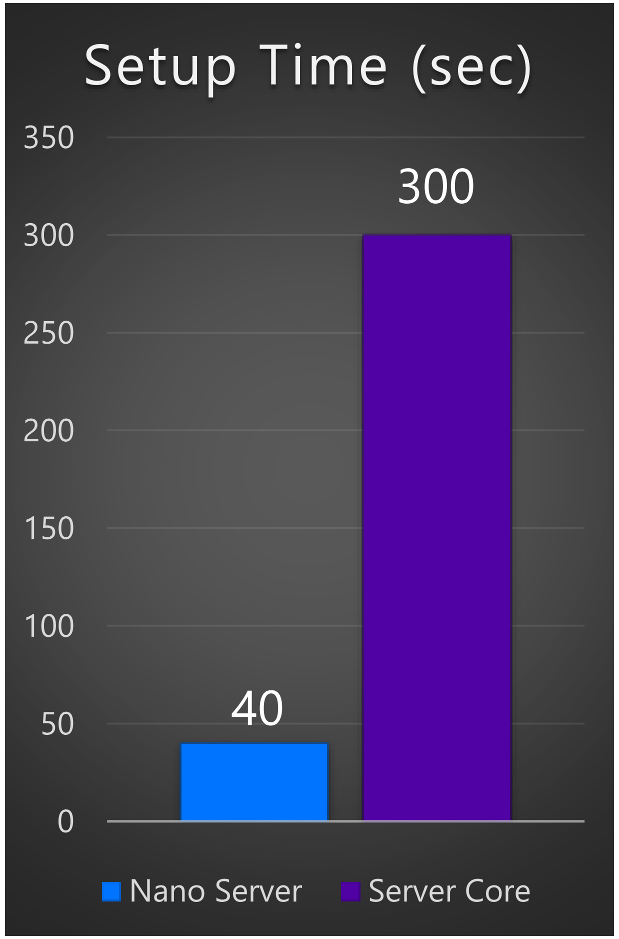
Preliminary Results



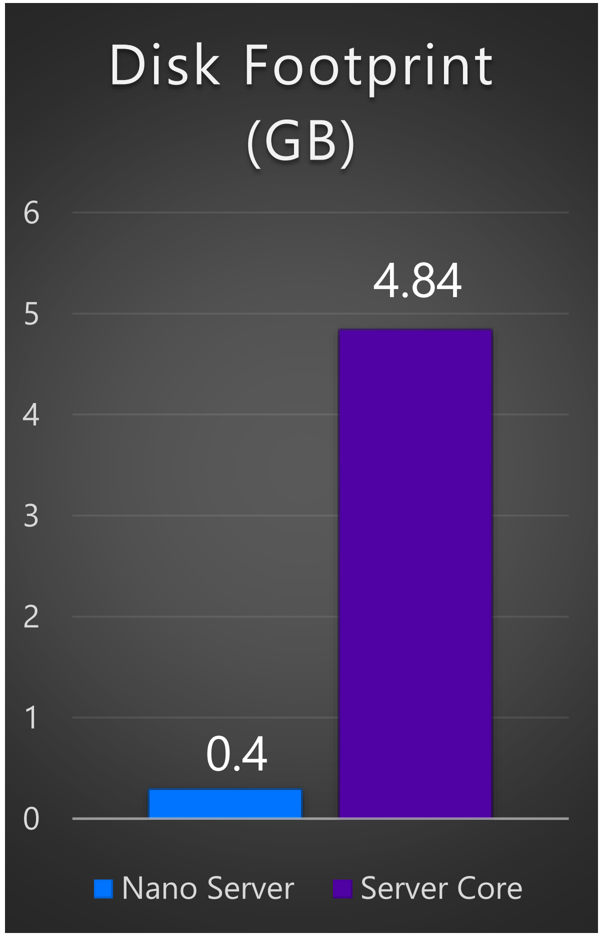
Critical Bulletins



Number of Reboots



Setup Time (sec)



Disk Footprint (GB)

Why Containers?

Containers empower application innovation

DevOps

Developers

Agility/productivity for app owners

Operations

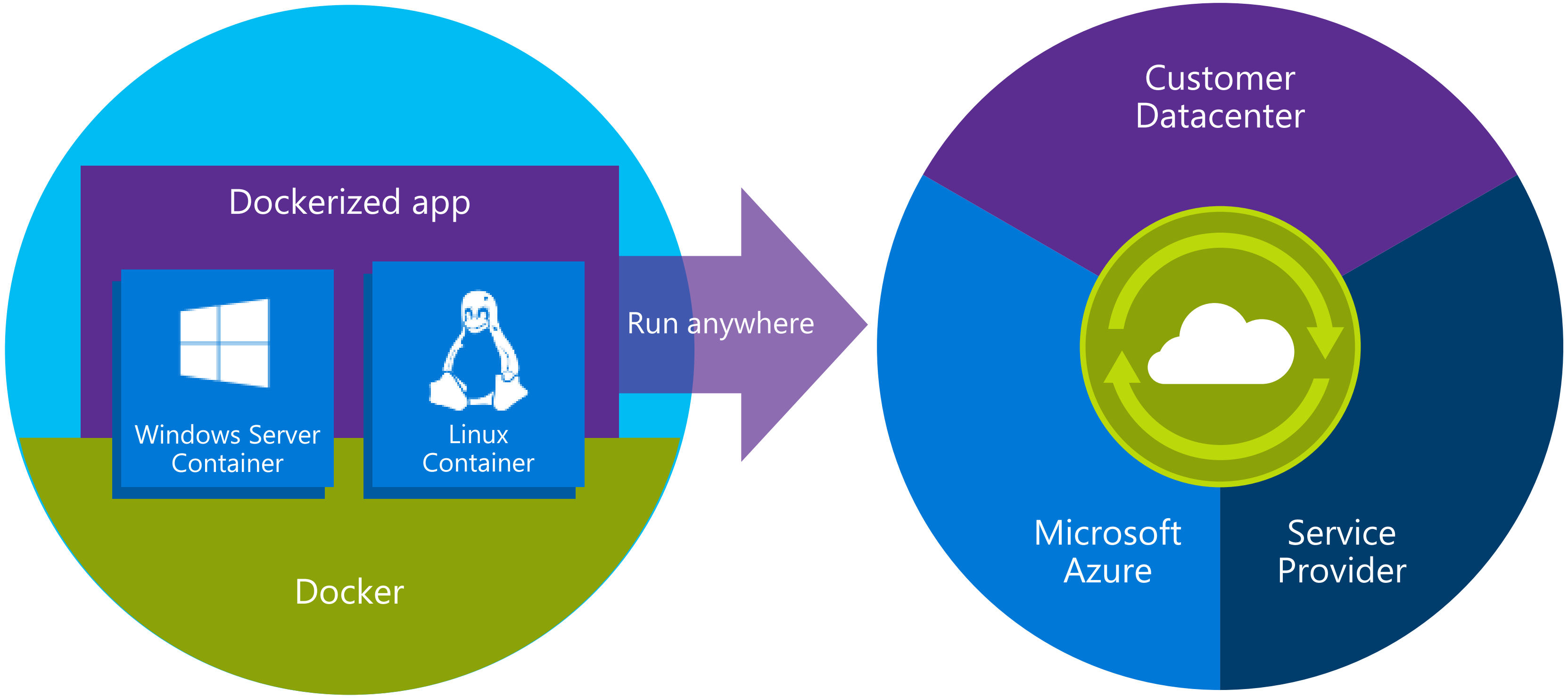
Flexibility and control for IT

Docker integration

Joint strategic investments to drive containers forward

Docker: An open source engine that automates the deployment of any application as a portable, self-sufficient container that can run almost anywhere.

Partnership: Enable the Docker client to manage multi-container applications using both Linux and Windows Server containers, regardless of the hosting environment or cloud provider.



Strategic investments {

Investments in upcoming Windows Server release

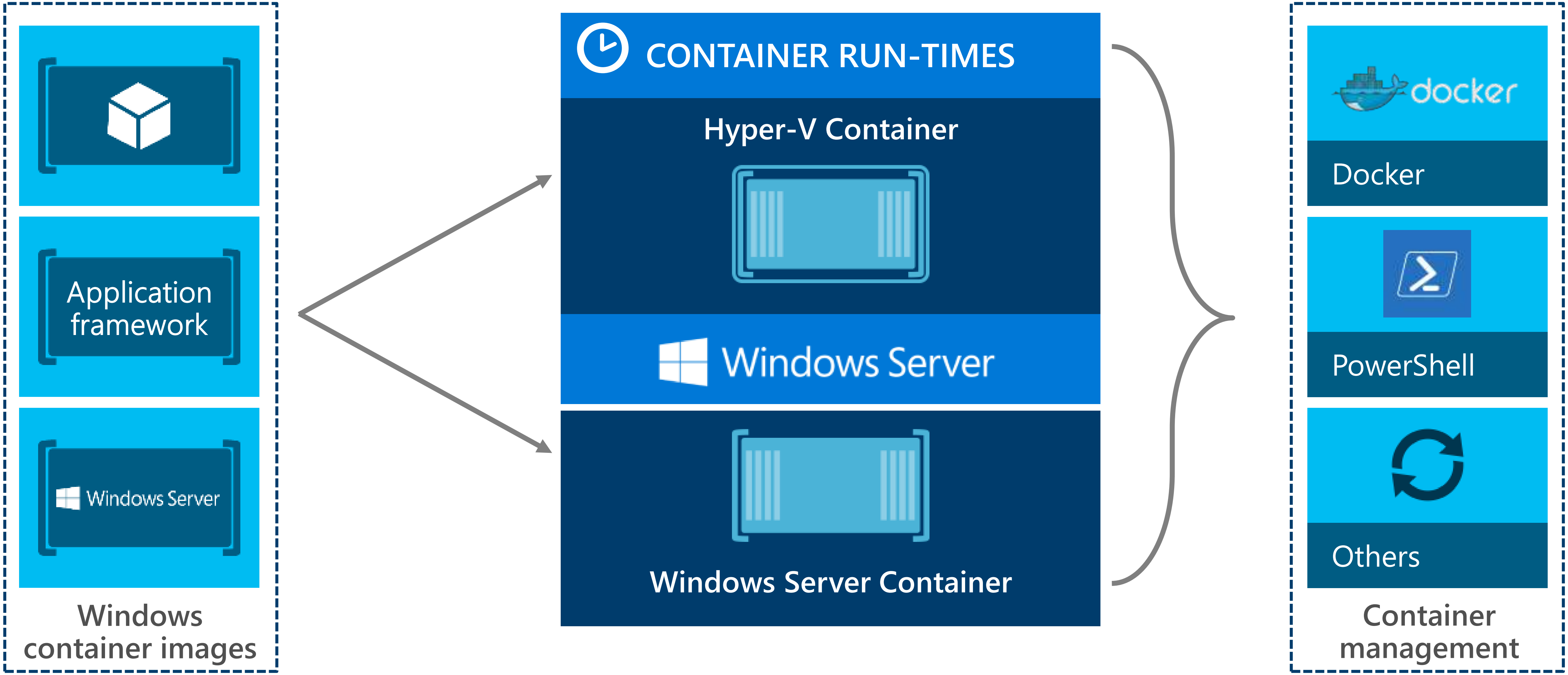
Open source development of the Docker Engine for Windows Server

Azure support for the Docker Swarm APIs

Federation of Docker Hub images into the Azure Gallery and Portal

Write once deploy anywhere

Modern app development with flexible isolation


Application framework
Windows Server
Windows container images
CONTAINER RUN-TIMES
Hyper-V Container
Windows Server
Windows Server Container
docker
Docker
PowerShell
Others
Container management