

# ZT Principles Applied to existing applications – DRAFT 0.2

ZT Working Group Project

# Goals

1. Compare guidance applying ZT principles to existing systems.
  1. NIST
  2. UK NCSC - Emerging
  3. EU NIS2 ENISA – Emerging
  4. Commercial
2. Establish recommendations for integration ZT and non-ZT systems.
3. Evaluate what resulting ZT + non-ZT Policy might looklike.

# The need for ZT + non-ZT guidance

CISA Observations from the CISA ZT Maturity Model

# CISA ZT MM (DRAFT)



## Zero Trust Maturity Model

Pre-decisional Draft

June 2021

Version 1.0

Cybersecurity and Infrastructure Security Agency  
Cybersecurity Division

[https://www.cisa.gov/sites/default/files/publications/CISA%20Zero%20Trust%20Maturity%20Model\\_Draft.pdf](https://www.cisa.gov/sites/default/files/publications/CISA%20Zero%20Trust%20Maturity%20Model_Draft.pdf)

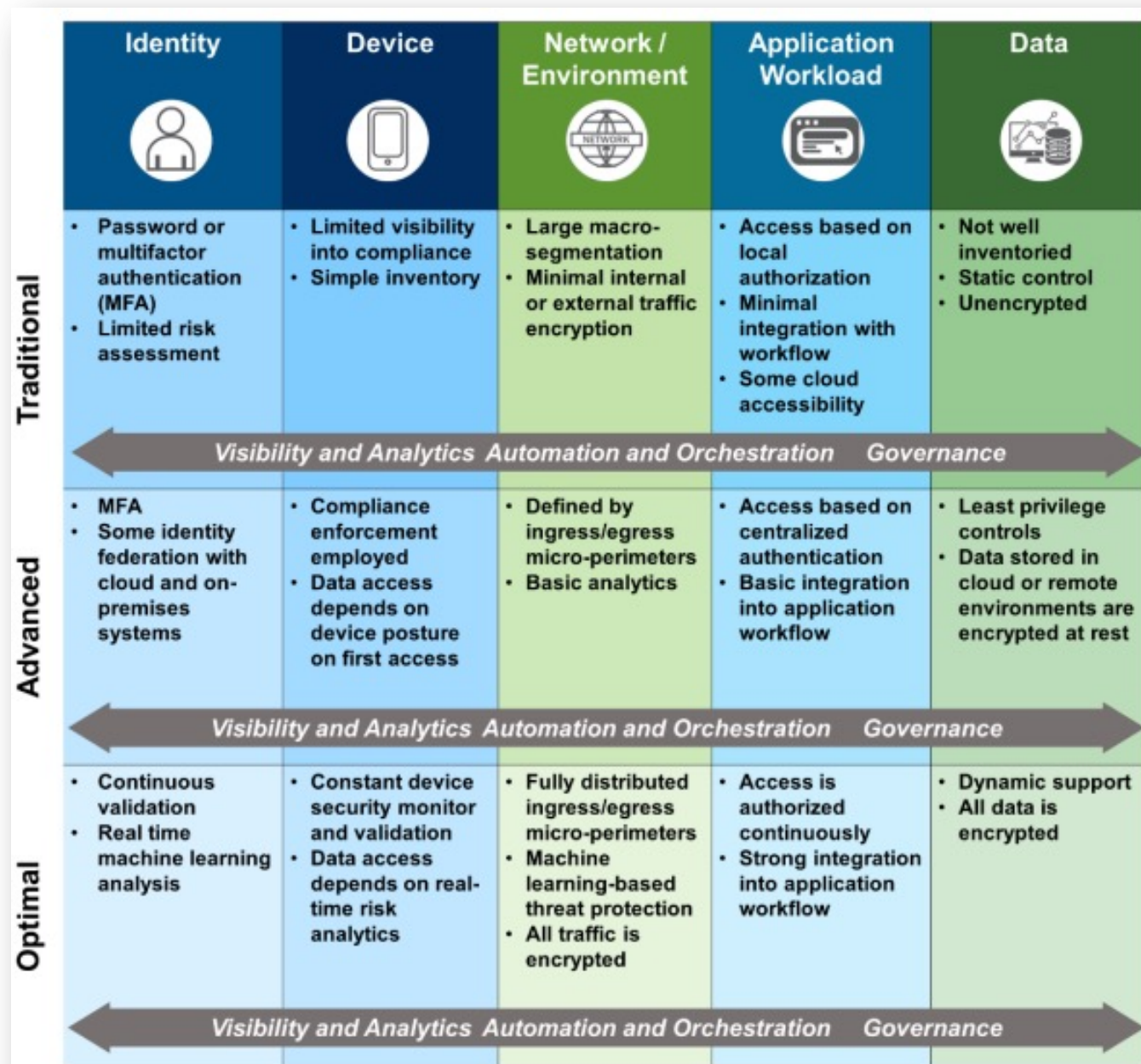


Figure 2: High-Level Zero Trust Maturity Model

# CISA MM Transitioning to ZT [Page 4]

- Transitioning to Zero Trust:
  1. Identify Actors on the Enterprise.
  2. Identify Assets Owned by the Enterprise.
  3. Identify Key Processes and Evaluate Risks Associated with Executing Process.
    - The Basis for Business Impact (Risk)
  4. Formulating Policies for the ZTA Candidate.
  5. Identifying Candidate Solutions.
  6. Initial Deployment and Monitoring

# Observations from CISA ZT MM Document

[Page ii]

- The path to zero trust is an incremental process that will take years to implement.
- Legacy infrastructure and systems may not support a zero trust implementation.

[Page 2] Zero trust presents a shift from a location-centric model to a more data-centric approach for fine-grained security controls between users, systems, data and assets that change over time; for these reasons, moving to a ZTA is non-trivial. This provides the visibility needed to support the development, implementation, enforcement, and evolution of security policies. More fundamentally, zero trust may require a change in an organization's philosophy and culture around cybersecurity. The path to zero trust is a journey that will take years to implement.

[Page 3] 6. Challenge The Federal Government faces several challenges in transitioning to ZTA. First, legacy systems rely on "implicit trust"; this concept conflicts with the core principle of adaptive evaluation of trust within a ZTA. Additionally, existing infrastructures are also built on implicit trust and must either be rebuilt or replaced. To rebuild or replace information technology (IT) infrastructure and mission systems requires a significant investment on the part of agencies. Lastly, there is no consensus on or formal adoption of a maturity model for ZTA. While proposals for maturity models have been put forth, current initiatives for kickstarting zero trust adoption are often focused on the network layer and do not present a holistic approach for transition

# NIST Application of ZT consistent security measures to existing systems in use.

NIST Security Measures for EO-Critical Software Use – ZT aligned, but not ZT complete

# NIST: Security Measures for EO-Critical Software Use

Aimed at near term implementation, in contrast to the years required to implement full zero trust and potential the replacement or development of existing systems.

- *(i) Within 60 days of the date of this order, the Secretary of Commerce acting through the Director of NIST, in consultation with the Secretary of Homeland Security acting through the Director of CISA and with the Director of OMB, shall publish guidance outlining security measures for critical software as defined in subsection (g) of this section, including applying practices of least privilege, network segmentation, and proper configuration.*
- *(j) Within 30 days of the issuance of the guidance described in subsection (i) of this section, the Director of OMB acting through the Administrator of the Office of Electronic Government within OMB shall take appropriate steps to require that agencies comply with such guidance.*
- *NIST has identified security measures that are fundamental for meeting these objectives. These “Security Measures for EO-Critical Software Use” are not intended to be comprehensive, nor are they intended to eliminate the need for other security measures that federal agencies implement as part of their existing requirements and cybersecurity programs. Agencies should continue their efforts to secure systems and networks that EO-critical software runs on and to manage cyber supply chain risk (see FAQ #4), as well as implement zero trust practices (see FAQ #5), which depend on the fundamental security measures. The intent of specifying these security measures is to assist agencies by defining a set of common security objectives for prioritizing the security measures that should be in place to protect EO-critical software use.*



# FAQ: EO-Critical SW Use

<https://www.nist.gov/itl/executive-order-improving-nations-cybersecurity/security-measures-eo-critical-software-use-1>

## **5.What is the relationship between this guidance and zero trust architecture?**

*Section 3 of the EO directs each federal agency to plan to implement zero trust architecture. All of the security measures for EO-critical software defined in this guidance are also components of a zero trust architecture, although by no means are they complete. Agencies developing plans for migrating to zero trust architecture can incorporate the security measures for EO-critical software use into those plans. For more information on zero trust architecture, see the following Federal Government resources:*

- DISA and NSA, [Department of Defense \(DOD\) Zero Trust Reference Architecture Version 1.0](#)
- NIST, [SP 800-207, Zero Trust Architecture](#)
- NSA, [Embracing a Zero Trust Security Model](#)

# Objective 1:

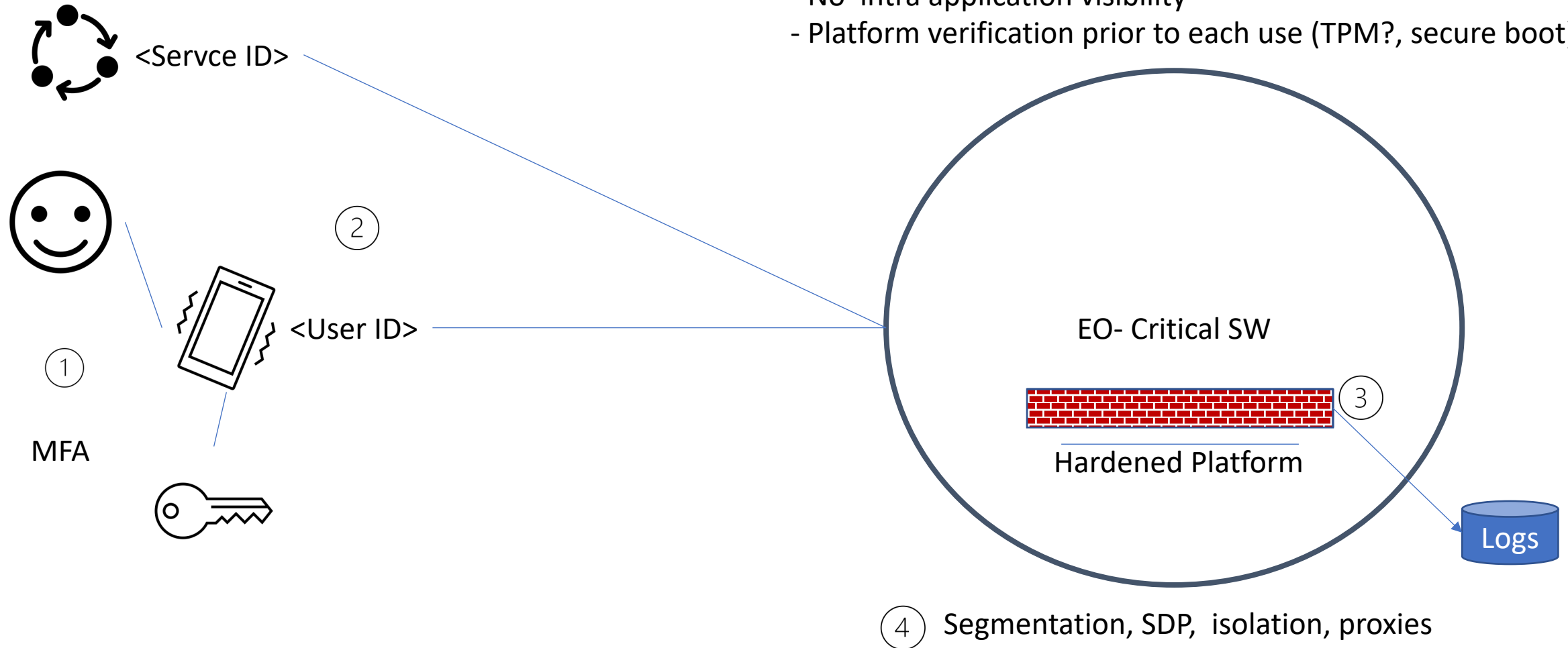
Objective 1: Protect EO-critical software and EO-critical software platforms from unauthorized access and usage.

- SM 1.1: Use multi-factor authentication that is verifier impersonation-resistant for all users and administrators of EO-critical software and EO-critical software platforms. (See FAQ #7.)
- SM 1.2: Uniquely identify and authenticate each service attempting to access EO-critical software or EO-critical software platforms.
- SM 1.3: Follow privileged access management principles for network-based administration of EO-critical software and EO-critical software platforms. Examples of possible implementations include using hardened platforms dedicated to administration and verified before each use, requiring unique identification of each administrator, and proxying and logging all administrative sessions to EO-critical software platforms.
- SM 1.4: Employ boundary protection techniques as appropriate to minimize direct access to EO-critical software, EO-critical software platforms, and associated data. Examples of such techniques include network segmentation, isolation, software-defined perimeters, and proxies.

# Objective 1

Differences from ZT:

- Coarser Granularity – bigger than a “container”
- Discovery of accessing Services by Observation
- No intra application visibility
- Platform verification prior to each use (TPM?, secure boot)

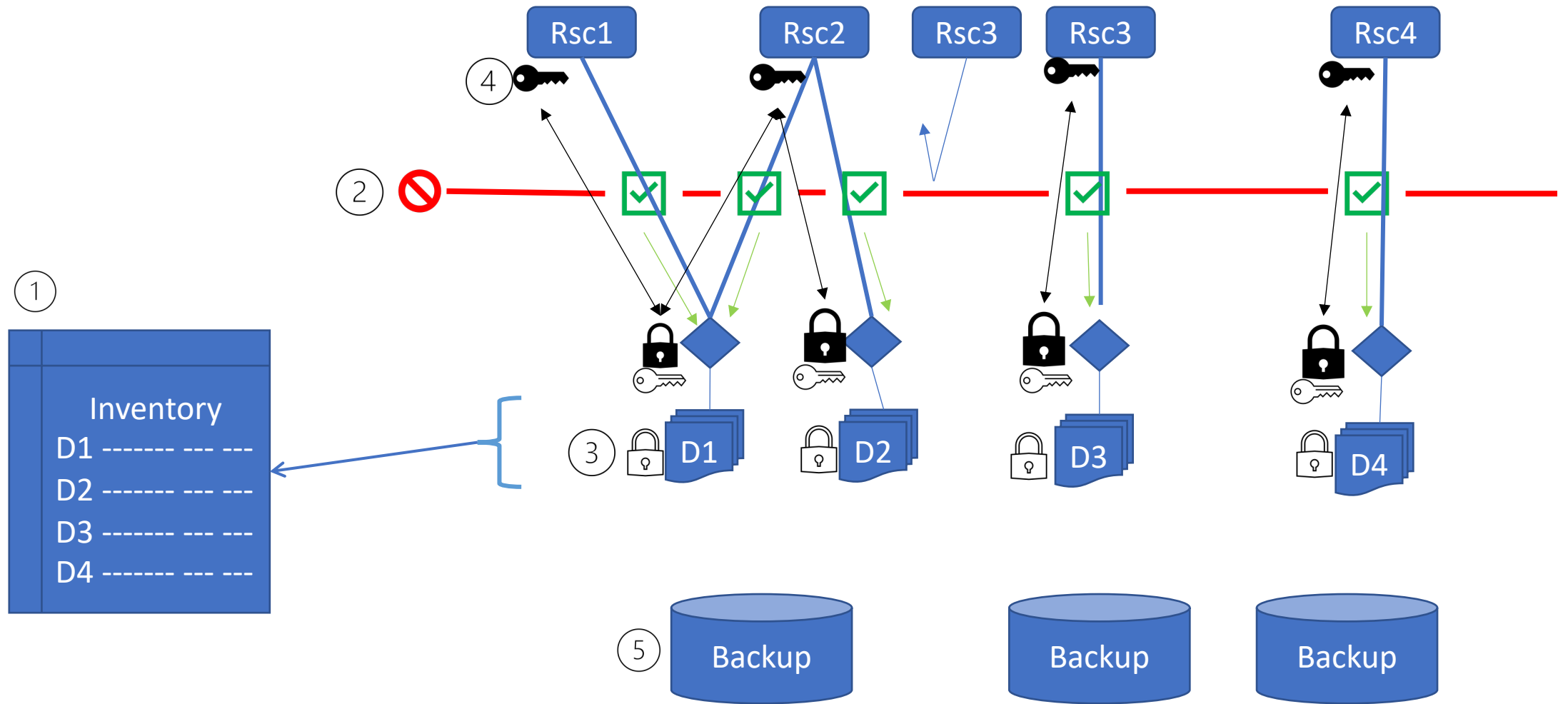


# Objective 2:

Objective 2: Protect the confidentiality, integrity, and availability of data used by EO-critical software and EO-critical software platforms. (See FAQ #6.)

- SM 2.1: Establish and maintain a data inventory for EO-critical software and EO-critical software platforms.
- SM 2.2: Use fine-grained access control for data and resources used by EO-critical software and EO-critical software platforms to enforce the principle of least privilege to the extent possible.
- SM 2.3: Protect data at rest by encrypting the sensitive data used by EO-critical software and EO-critical software platforms consistent with NIST's cryptographic standards.
- SM 2.4: Protect data in transit by using mutual authentication whenever feasible and by encrypting sensitive data communications for EO-critical software and EO-critical software platforms consistent with NIST's cryptographic standards.
- SM 2.5: Back up data, exercise backup restoration, and be prepared to recover data used by EO-critical software and EO-critical software platforms at any time from backups.\\

# Objective 2 Diagram



# Objective 3:

Objective 3: Identify and maintain EO-critical software platforms and the software deployed to those platforms to protect the EO-critical software from exploitation.

- SM 3.1: Establish and maintain a software inventory for all platforms running EO-critical software and all software (both EO-critical and non-EO-critical) deployed to each platform.
- SM 3.2: Use patch management practices to maintain EO-critical software platforms and all software deployed to those platforms. Practices include: rapidly identify, document, and mitigate known vulnerabilities
- SM 3.3: Use configuration management practices to maintain EO-critical software platforms and all software deployed to those platforms. Practices include: (hardened security configurations enforce the principles of least privilege, separation of duties, and least functionality)

# Objective 4:

Objective 4: Quickly detect, respond to, and recover from threats and incidents involving EO-critical software and EO-critical software platforms.

- SM 4.1: Configure logging to record the necessary information about security events involving EO-critical software platforms and all software running on those platforms.
- SM 4.2: Continuously monitor the security of EO-critical software platforms and all software running on those platforms.
- SM 4.3: Employ endpoint security protection on EO-critical software platforms to protect the platforms and all software running on them. Capabilities include:
  - protecting the software, data, and platform by identifying, reviewing, and minimizing the attack surface and exposure to known threats.
  - permitting only verified software to execute (e.g., file integrity verification, signed executables, allow listing)
- SM 4.4: Employ network security protection to monitor the network traffic to and from EO-critical software platforms to protect the platforms and their software using networks. Capabilities include:
- SM 4.5: Train all security operations personnel and incident response team members, based on their roles and responsibilities, on how to handle incidents involving EO-critical software or EO-critical software platforms.

# Objective 5

Objective 5: Strengthen the understanding and performance of humans' actions that foster the security of EO-critical software and EO-critical software platforms.

- SM 5.1: Train all users of EO-critical software, based on their roles and responsibilities, on how to securely use the software and the EO-critical software platforms.
- SM 5.2: Train all administrators of EO-critical software and EO-critical software platforms, based on their roles and responsibilities, on how to securely administer the software and/or platforms.
- SM 5.3: Conduct frequent awareness activities to reinforce the training for all users and administrators of EO-critical software and platforms, and to measure the training's effectiveness for continuous improvement purposes.



# Objective 2

# Objective 3

# Objective 4

# Objective 5

# Commercial Example - Microsoft

Focus: Zero Trust Model helps by:

- Applying controls and technologies to discover Shadow IT.

- Ensuring appropriate in-app permissions.

- Limiting access based on real-time analytics.

- Monitoring for abnormal behavior.

- Controlling user actions.

- Validating secure configuration options.

Initial Deployment Objectives:

- I. Gain visibility into the activities and data in your applications by connecting them via APIs.

- II. Discover and control the use of shadow IT.

- III. Protect sensitive information and activities automatically by implementing policies.

Additional deployment objectives:

- IV. Deploy adaptive access and session controls for all apps.

- V. Strengthen protection against cyber threats and rogue apps.

- VI. Assess the security posture of your cloud environments

<https://docs.microsoft.com/en-us/security/zero-trust/deploy/applications>

# Next Steps – In Process

Feel free to adjust, correct refine, ...

# Next Steps

- Diagram for each Objective
  - Is the diagram for Objective 1 an adequate first iteration for each objective?
- Identify differences from more conformant ZT
- Commercial examples of applying ZT Principles to existing systems
  - Microsoft
  - AWS
  - Google
  - \* <other group contributors may want to consider summarizing these or others>

# Questions – for consideration by the broader group

- Does emerging guidance (non-ZT) provide abstractions for aligning postures for integrated ZT and non-ZT ?
  - Isolation
  - Authentication & verification
  - Trust-ability of the platform
  - Completeness of the User, Service universe
- Are there limits to integration that should be considered?
  - Directional limitations?
  - Restrictions on change (to address supply chain absent devops )?
  - Policy on detection/nonconformance in non-ZT domain?