Is Your Compute Environment Holding You Back? A DIB Guide for the Acquisition Community

To enable software to provide a competitive advantage to the warfighter, DoD must adopt a strategy for rapidly transitioning DoD IT to current industry standards. This modernization agenda should include providing distributed databases and abundant computing power; making bandwidth available as a platform; integrating mobile technologies; and developing DoD platforms for downloading applications. This document outlines compute and infrastructure capabilities that should be available to DoD programmers (and contractors) who are developing software for national defense. The capabilities include:

1. **Scalable compute.** Access to computing resources should never be a limiting factor when developing code. Modern cloud environments provide mechanisms to provide any developer with a powerful computing environment that can easily scale with the needs of an individual programmer, a product development team, or an entire enterprise.

2. **Containerization.** Container technology provides sandbox environments in which to test new software without exposing the larger system to the new code. It “packages up” an application with all of the operating system services required for executing the application and allowing that application to run in a virtualized environment. Containers allow isolation of components (that communicate with each other through well-defined channels) and provide a way to “freeze” a software configuration of an application without freezing the underlying physical hardware and operating system.

3. **Continuous integration/continuous delivery (CI/CD) pipeline (DevSecOps platform).** A platform that provides the CI/CD pipeline is used for automated testing, security, and deployment. This includes license access for security tools and a centralized artifacts repository with tools, databases, and a base operating system (OS) with an existing authorization to operate (ATO).

4. **Infrastructure as code: automated configuration, updating, distribution, and recovery management.** Manual configuration management of operating systems and middleware platforms leads to inconsistencies in fielded systems and drives up the operating costs due to the labor hours required for systems administration. Modern software processes avoid this by implementing “infrastructure as code,” which replaces manual processes for provisioning infrastructure with automated processes that use machine-readable definition files to manage and provision containers, virtual machines, networking, and other components. Adopting infrastructure as code and software distribution tools in a standardized way streamlines uniformity of deployment and testing of changes, which are both vital to realizing the benefits of agile development processes.

5. **Federated identity management and authentication backend with common log file management and analysis.** Common identity management across military, government, and contractors greatly simplifies the assignment of permissions for accessing information across multiple systems and allows rapid and accurate auditing of

code. The ability to audit access to information across multiple systems enables the detection of inappropriate access to information, and can be used to develop the patterns of life that are essential for proper threat analysis. Common identity management can ease the integration of multi-factor authentication across servers, desktops, and mobile devices. Along with public key infrastructure (PKI) integration, it allows verification of both the service being accessed by the user and the user accessing information from the service.

6. **Firewall configuration and network access control lists.** Having a common set of OS and application configurations allows network access control not just through network equipment, but at the server itself. Pruning unnecessary services and forcing information transfer only through intentional interfaces reduce the attack surface and make servers more resilient against penetration. Server-to-server communication can be encrypted to protect from network interception and authenticated so that software services can only communicate with authorized software elements.

7. **Client software.** Remote login through remote desktop access is common throughout DoD. This greatly increases the difficulty of integrating mobile platforms and of permitting embedded devices to access vital information, especially from the field. It also complicates uniform identity management and multi-factor verification, which is key to securing information. By moving to web client access mobile integration - and development - is greatly eased. It also becomes possible to leverage industry innovation, as this is where the commercial sector is heading for all interactions.

8. **Common information assurance (IA) profiles.** Information assurance (IA) for DoD systems is complex, difficult, and not yet well-architected. Test, certification, and IA are almost always linear “tailgate” processes instead of being integrated into a continuous delivery cycle. Common IA profiles integrated into the development environment and part of the development system architecture are less likely to have bugs than customized and add-on solutions.

Desired State with Examples

Effective use of software requires sufficient resources for computing, storage, and communications. Software development teams must be provided with abundant compute, storage, and bandwidth to enable rapid creation, scaling, and optimization of software products.

Modern cloud computing services provide such environments and are widely available for government use. In its visits to DoD programs, the DIB Software Acquisition and Practices (SWAP) team has observed many programs that are regenerating computing infrastructure on their own—often in a highly non-optimal way—and typically due to constraints (or perceived constraints) created by government statutes, regulations, and culture. This approach results in situations where compute capability does not scale with needs; operating systems cannot be upgraded without upgrading applications; applications cannot be upgraded without updating the operating systems; and any change requires a complete information assurance recertification.

Compute platforms are thus “frozen” at a point established early in the program life cycle, and development teams are unable to take advantage of new tools and new approaches as they become available. The DIB SWAP team has noted a general lack of good tools for profiling code, maintaining access and change logs, and providing uniform identity management, even though the DoD has system-wide credentials through Common Access Control (CAC) cards.

It would be highly beneficial to create common frameworks and/or a common set of platforms that provide developers with a streamlined or pre-approved Authority to Operate (ATO). Use of these pre-approved platforms should not be mandated, but they create cost and time incentives by enabling more consolidated platforms. DoD could make use of emerging government cloud computing platforms or achieve similar consolidation within a DoD-owned data center (hybrid cloud). DoD should move swiftly from a legacy data center approach to a cloud-based model, while taking into account the lessons learned and tools and services available from commercial industry, with assumed hardware and operating system updates every 3-5 years.

Warning signs

Some indicators that you may have screwed up your compute environment include:

- Your programmers are using tools that are less effective than what they used in school
- The headcount needed to support the system grows linearly with the number of servers or instances
- You need system managers deployed with hardware at field locations because it is impossible to configure new instances without high skill local support
- You have older than current versions of operating systems or vendor software because it is too hard to test or validate changes
- Unit costs for compute, network transport and storage are not declining, or are not measurable to be determined
- Logging in via remote desktop is the normal way to access an information service
- You depend on network firewalls to secure your compute resource from unauthorized access
- You depend on hardware encryptors to keep your data safe from interception
- You have to purge data on a regular basis to avoid running out of storage
- Compute tasks are taking the same or longer time to run than they did when the system was first fielded
- Equipment or software is in use that has been “end of lifed” by the vendor and no longer has mainstream support
- It takes significant work to find out who accessed a given set of files or resources over a reasonable period of time
- No one knows what part of the system is consuming the most resources or what code should be refactored for optimization
- Multifactor authentication is not being used
- You cannot execute a disaster recovery exercise where a current backup up of a system cannot be brought online on different hardware in less than a day

Getting It Right

These capabilities should be available to all DoD programmers and contractors developing software for national defense:

Scalable compute

- Modern compute architectures
- Environments that make transitions across cloud and local services easy
- Graphics Processing Unit (GPU)- and ML-optimized compute nodes available for specialized tasks
- Standardized storage elements and ability to expand volumes and distribute them based on performance needs
- Standardized network switching options with centralized image control
- Property management tagging—no equipment can be placed in a data center without being tagged for inventory and tracked for End of Life support from vendors
- Supply chain tracking for all compute elements

Containerization

- Software deployment against standard profile OS image
- Containers can be moved from physical to cloud-based infrastructure and vice versa
- Applications and services run in containers and expand or contract as needed
- OS updates separated from application container updates
- Centralized OS patch validation and testing
- Containers can be scaled massively horizontally
- Containers are stateless and can be restarted without impact
- Configuration management for deployment and audit

Continuous integration/continuous delivery (CI/CD) pipeline (DevSecOps platform)

- Select, certify, and package best of breed development tools and services
- Can be leveraged across DoD Services as a turnkey solution
- Develop standard suite of configurable and interoperable cybersecurity capabilities
- Provide onboarding and support for adoption of Agile and DevSecOps
- Develop best-practices, training, and support for pathfinding and related activities
- Build capability to deliver a Software Platform to the Defense Enterprise Cloud Environment
- Self-service portal to selectively configure and deliver software toolkit with pre-configured cybersecurity capabilities

Infrastructure as code: automated configuration, updating, distribution and recovery management

- Ability to test changes against dev environments
- Standardized profiling tools for performance measurement
- Centralized push of patches and updates with ability for rapid rollback
- Auditing and revision control framework to ensure proper code is deployed and running
- Ability to inject faults and test for failover in standardized ways

- Disaster recovery testing and failover evaluation
- Utilization tracking and performance management utilities to predict resource crunches
- Standardized OS patch and distribution repositories
- Validation tools to detect manual changes to OS or application containers with alerting and reporting

Federated identity management and authentication backend with common log file management and analysis

- Common identity management across all DoD and contractors
- Common multifactor backends for authentication of all users along with integration of LDAP/Radius/DNS or active directory services
- Integrated PKI services and tools for automated certificate installation and updating
- Common DRM modules that span domains between DoD/contractors and vendor facilities that can protect, audit and control documents, files, and key information. All encrypted at rest, even for plain text files.
- Useful for debugging and postmortem analysis
- Develop patterns of life to flag unusual activity by users or processes
- Automated escalation to defensive cyber teams

Firewall configuration and network access control lists

- Default configuration for containers is no access
- Profiles for minimal amounts of ports and services being open/run
- All network communications are encrypted and authenticated, even on the same server/container

Client software

- Web-based access the norm, from desktops/laptops as well as mobile devices
- Remote login used as a last resort - not as the default
- Security technical implementation guides (STIGs) for browsers and plugins, as well as common identity management at the browser interface (browsers authenticate to servers as well as servers authenticating to browsers)
- Minimal state kept on local hardware - purged at end of session

Common information assurance (IA) profiles

- Enforces data encrypted in flight and at rest
- Software versions across DoD with automated testing
- Application lockdowns at the system level so only authorized applications can run on configured systems
- “Makefile” to build configurations from scratch from base images in standardized approved configurations
- Use of audit tools to detect spillage and aid in remediation (assisted via DRM