truMLS®

# Containerized Digital Engineering Applications through Multilevel Secure Architectures

This presentation is UNCLASSIFIED

October 25, 2023

GENERAL DYNAMICS
Mission Systems

PRI-2310-0032

# Introductions

**Joyce Fai**
truMLS Portfolio
Technical Lead

**Scott Devitt**
truMLS Products
Solution Architect

**Nate Nadeau**
truMLS Software
Engineer

# Digital Engineering (DE) through Multilevel Secure Architectures

- Techniques for Multilevel DE Environment
  - Polyinstantiation
  - Multilevel Secure Containerization

- Design Patterns for Multilevel DE Environment
  - Pattern 1: Use Multilevel Share for MATLAB Collaboration
  - Pattern 2: Use Multilevel Secure MagicDraw/TWC to Publish and Review
  - Pattern 3: Use Multilevel Share to Collaborate and Share DE Labelled Models
  - Pattern 4: Consolidate Classified CI/CD Pipelines with MLS
  - Pattern 5: Jira and Confluence in Multilevel Secure Containers

All classification markings within this presentation are representative, unclassified, and for demonstration purposes only.

# What is Multilevel Security (MLS)?

- Multilevel Security is the application of computer systems, operating systems, storage systems and applications to simultaneously process information at different security levels for users with different authorizations/clearances and need-to-know.

- Multilevel Security provides:
  - Isolation of processes, users, and data at different sensitivity levels (e.g., classification, compartments, projects, or proprietary information)
  - Isolation of connections to multiple networks at different sensitivity levels
  - Isolation of users with different clearances using the system simultaneously at different sensitivity levels
  - Isolation of resources allocated by processes and services operating and serving different sensitivity levels

# Protection Levels in the USG Communities

PL2: “Single Level/System High” Users and data at the same classification level and same access;

- DAC (RBAC)

PL3: “Need to Know” Users and data at the same classification level, different accesses/compartments;

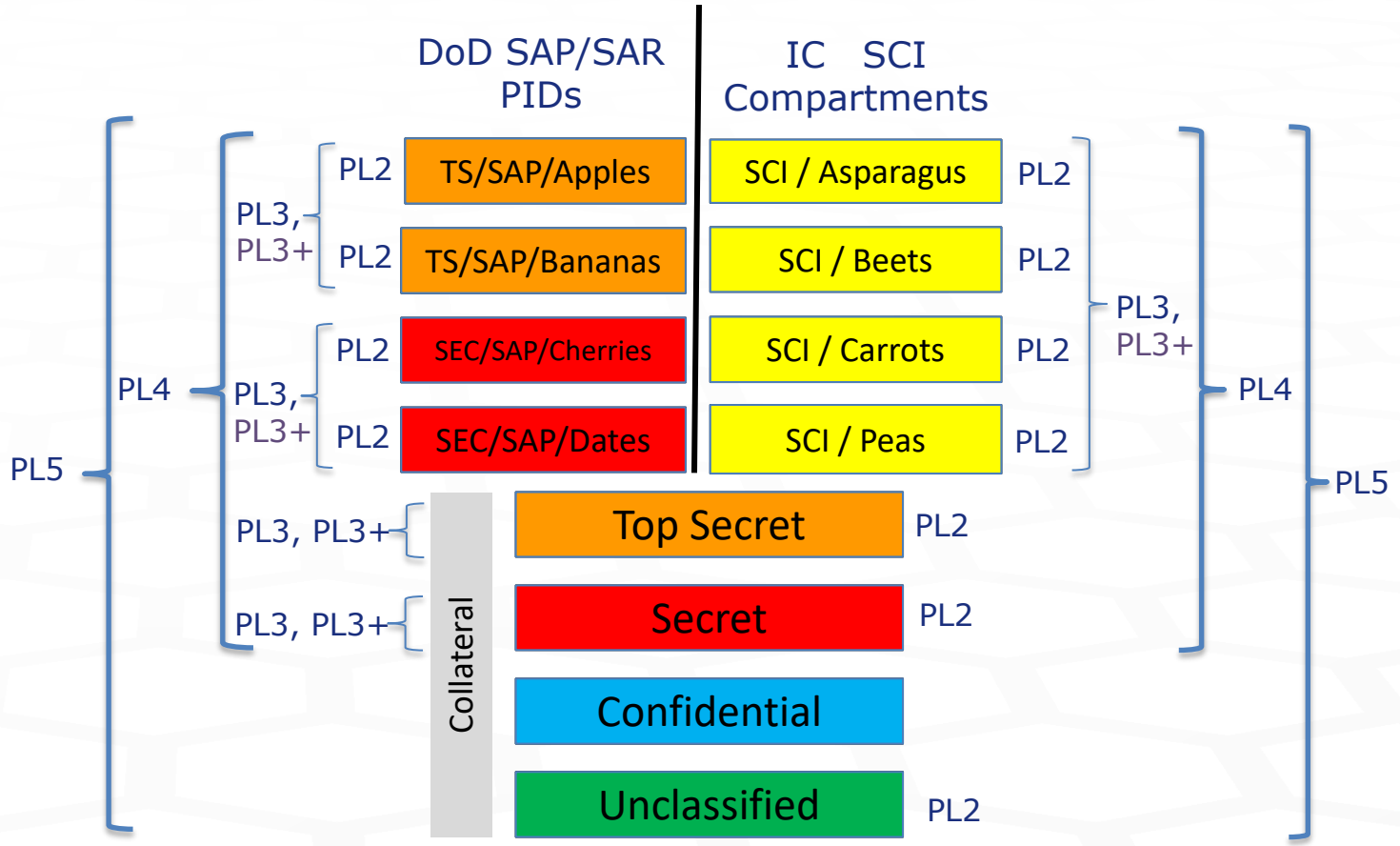
- DAC+ (RBAC)

PL3+: “Need to Know” Users and data at the same classification level, different accesses/ compartments; different organizations (e.g., AF, Army, Navy) and stronger separation

- DAC+ (RBAC) + MAC (RBAC, TCB)

PL4: “MLS” Users and data have different classification levels and different accesses;

- DAC & MAC (RBAC, TCB)



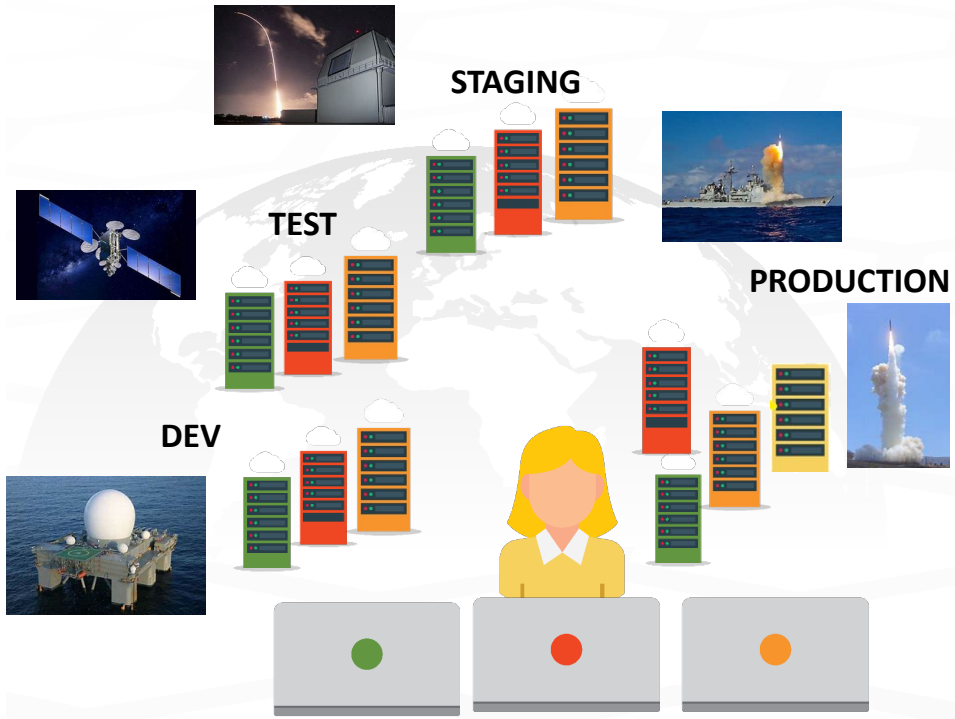

- MAC is not restricted to SAP or multiple classification levels
- MAC is suitable for single level collateral networks that need to protect program specific information, industry partner sensitive and proprietary information.

# DE Modernized with Multilevel Security

## *Challenge: Network-centric Approach*

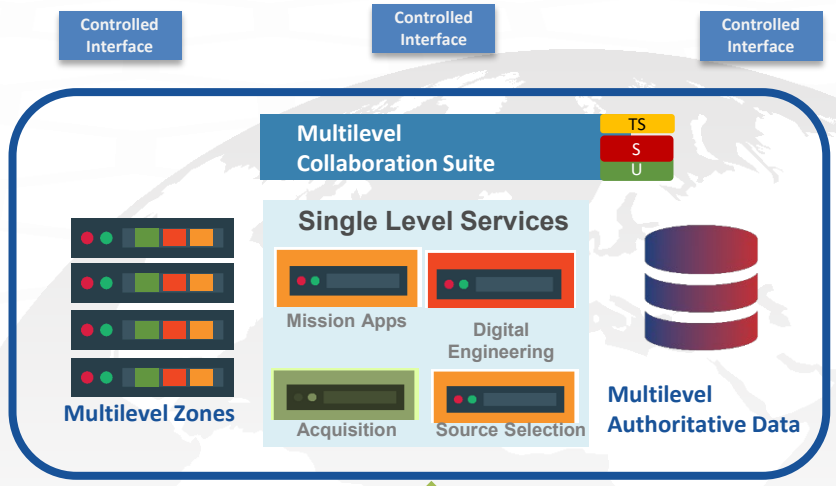
*Multiple Classifications, Multiple Networks, Traditional Data Silos*

- Silos prevent ease of collaboration, analysis
- Too many systems to access
- Inefficient user workflow
- Data tends to drift upward to “system high”
- Very high infrastructure costs




## *Data-centric Enterprise Multilevel Secure Approach*

*One or more Classifications, Compartments, Programs; One Network*




**Shared single level and multilevel services in MLS core**

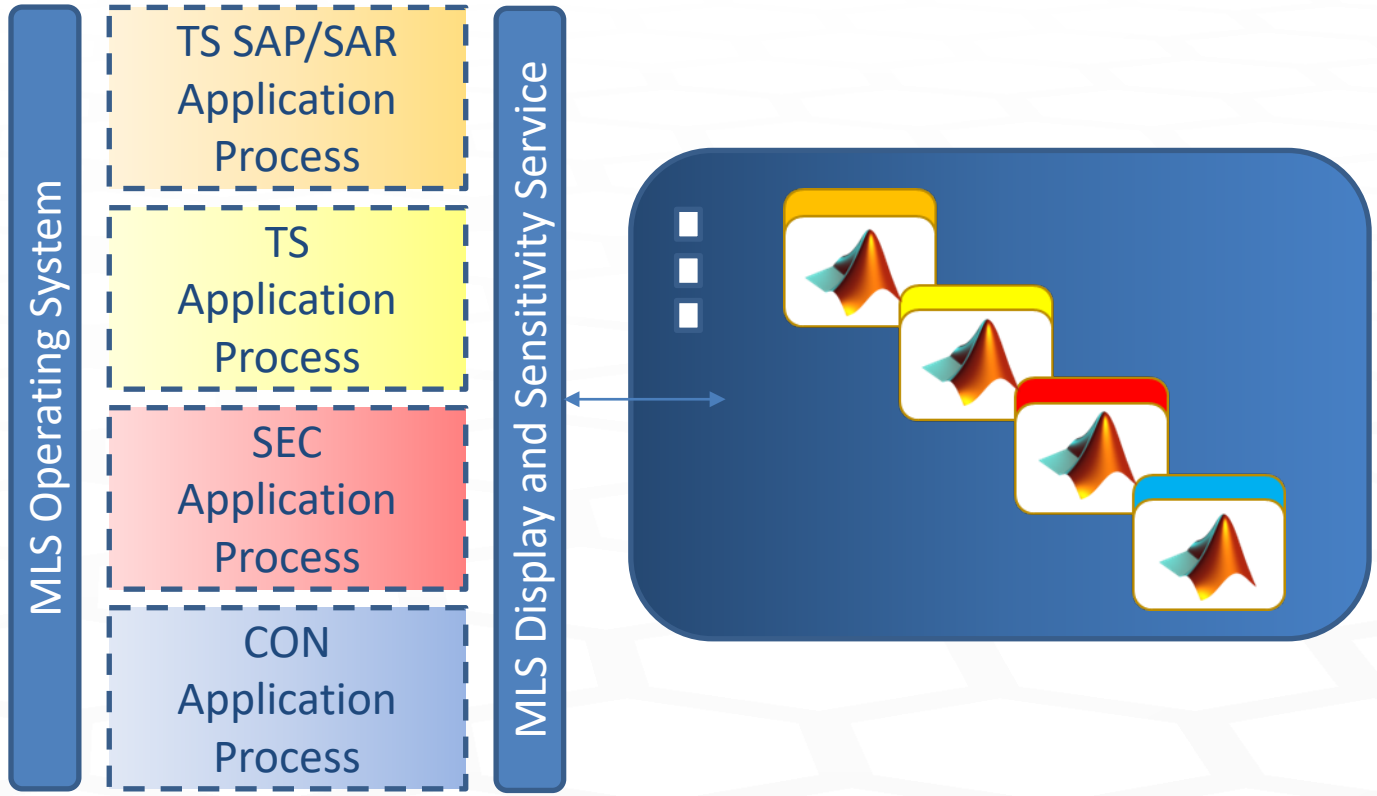
- No more moving complex digital engineering models and code between networks reducing delay
- **Reduced infrastructure footprint**
- Interoperability with single level commercial applications
- Avoid over-classification and system high limitation by labeling at the correct level and providing access based on a user’s credentials.
- Maximize workforce to mitigate shortfall of program cleared personnel

**Single pane of glass**

- Iteratively design models at multiple security levels and/or multiple domains from single desktop
- Develop code at the appropriate security level and have access to code repositories at different security levels

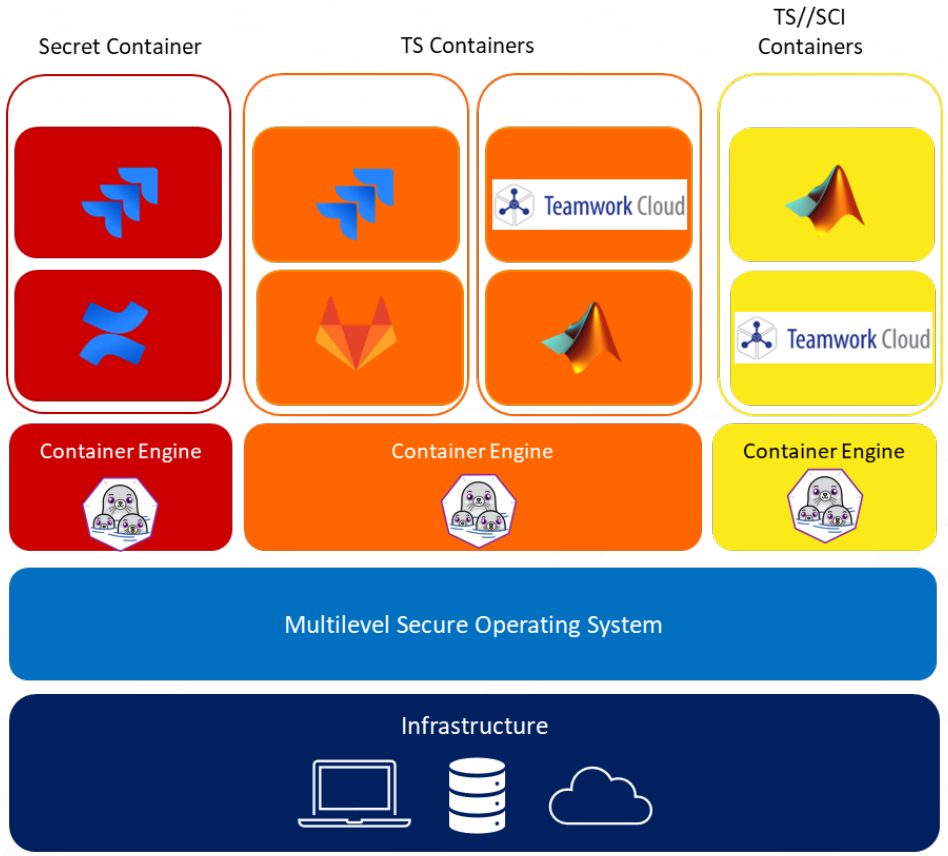
# DE via Polyinstantiation

- Applications that support Linux can run on an MLS OS
- Configure the MLS OS to run 3rd party DE applications at specific sensitivity levels
- All processes and files are isolated by MAC label segregation
- Linux-based applications with host-based licensing yields cost savings when running in polyinstantiated mode



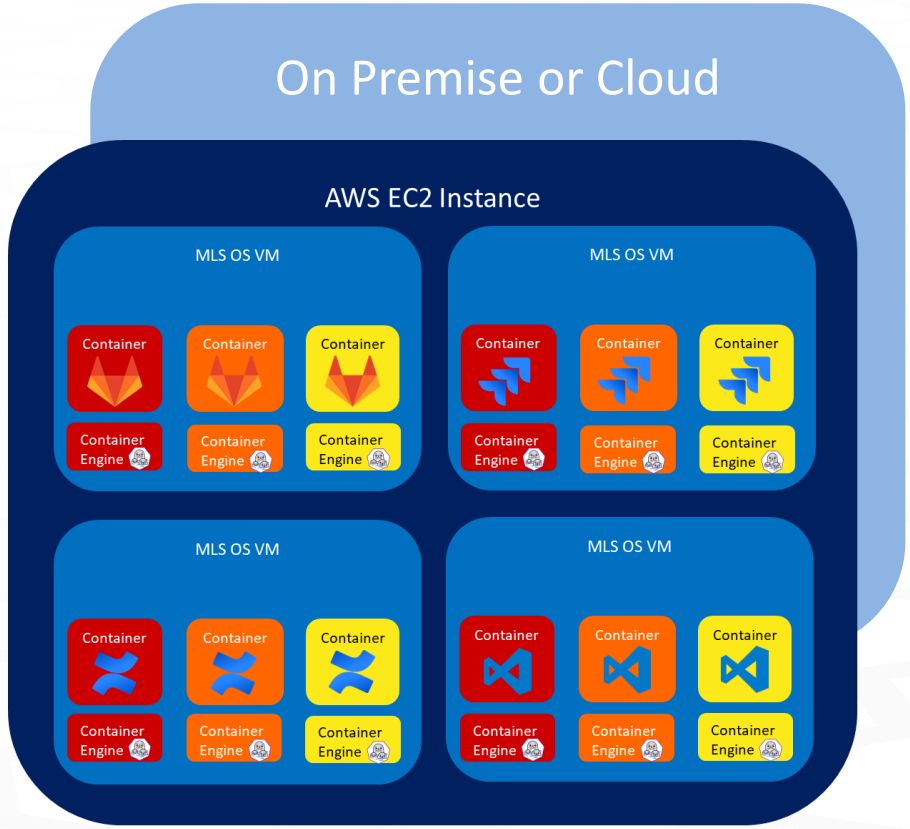

**Polyinstantiation**: multiple instances of the same application are invoked with separate resources to allow a subject with minimal privileges to create data that is automatically segregated by sensitivity

# Multilevel Secure Containerization Framework and Deployment




**Flexibility in Deployment:**

- Deploy MLS Secure OS and containers in the cloud with Infrastructure as Code (IaC)
- Deploy MLS Secure OS and containers on premise using automation scripts

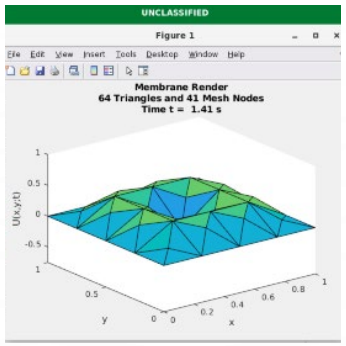


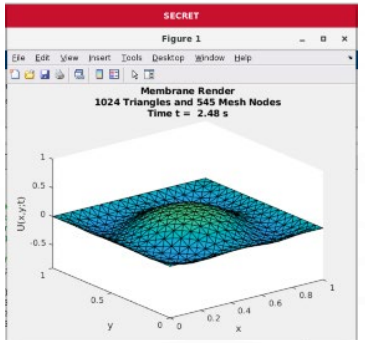

- Polyinstantiated container engine
- One instance per sensitivity level
- Containers isolated at multiple levels

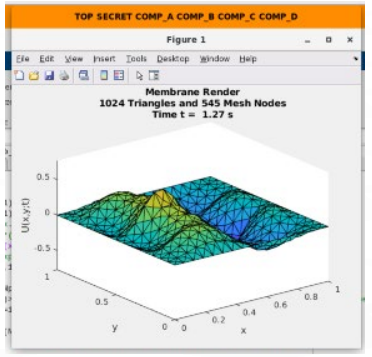
**Containerization**: leverage single level, third party tools in a multilevel environment providing customers flexibility in deployment while maintaining rigorous data segregation

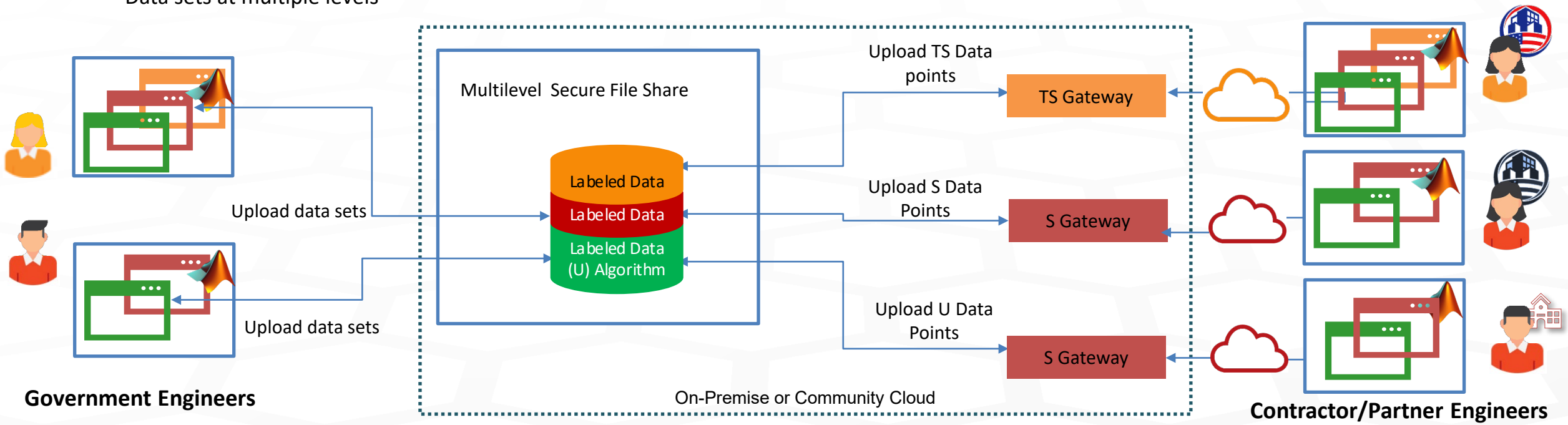
# Pattern 1: Use Multilevel Share for MATLAB Collaboration

1. Polyinstantiate MATLAB
   - Engineering models at multiple levels from single pane of glass
2. Deploy multilevel file share to store labelled data
   - Common unclassified algorithm
   - Data sets at multiple levels
3. Limit access to resource based on sensitivity of the resource and authorization of the user [ZT]
4. Enforce separation and labelling at the network level [ZT]



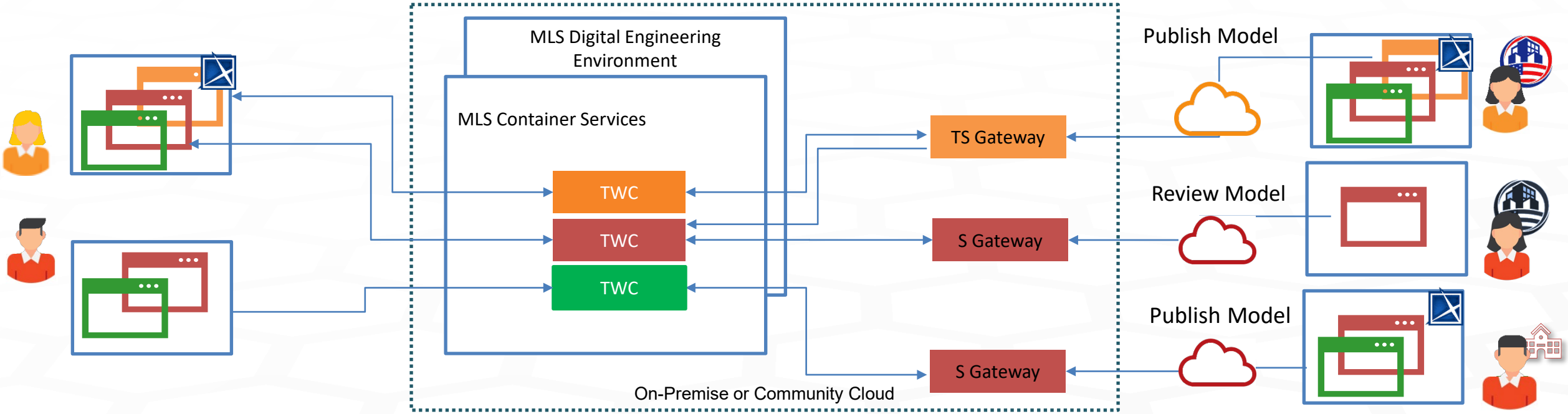



# Pattern 2: Use Multilevel Secure MagicDraw/TWC to Publish and Review

1. Polyinstantiate MagicDraw
   - Authors update models at multiple levels from single pane of glass
2. Containerize and Deploy TWC at Multiple Levels
   - Single infrastructure
3. Enforce separation at network level [ZT]
4. Control what network resources can be used and ensures secure single level network collaboration [ZT]



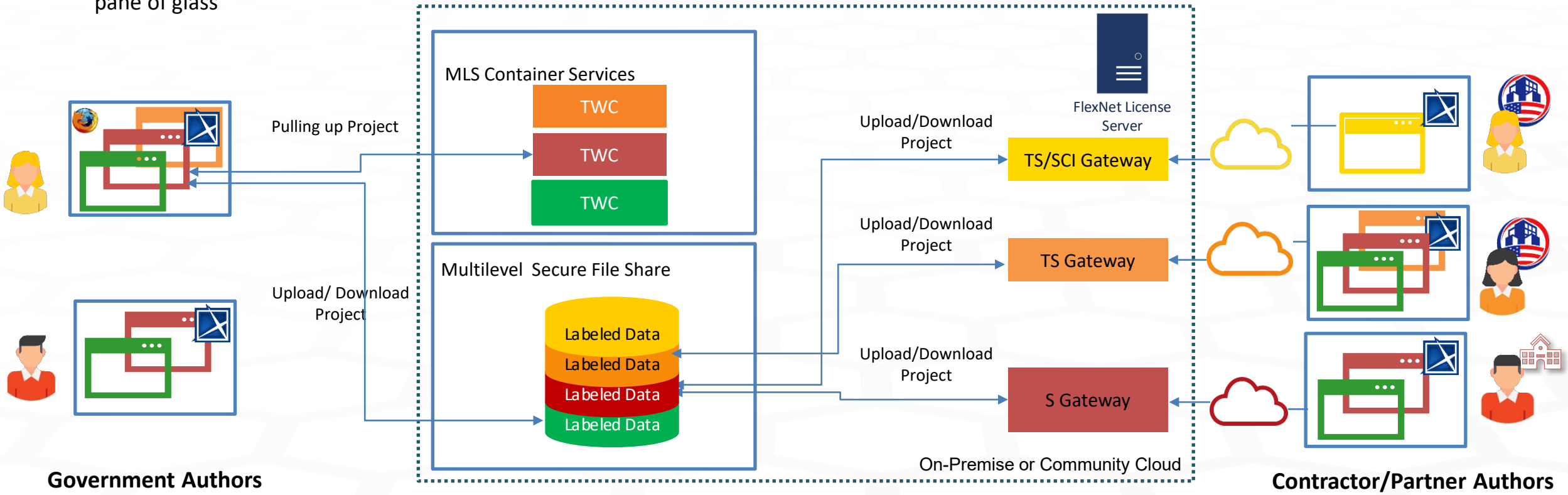

**Authors and Reviewers - Single Pane of Glass**

5. Limit access to resource based on sensitivity of the resource and authorization of the user [ZT]

**Authors and Reviewers – Single Pane of Glass**

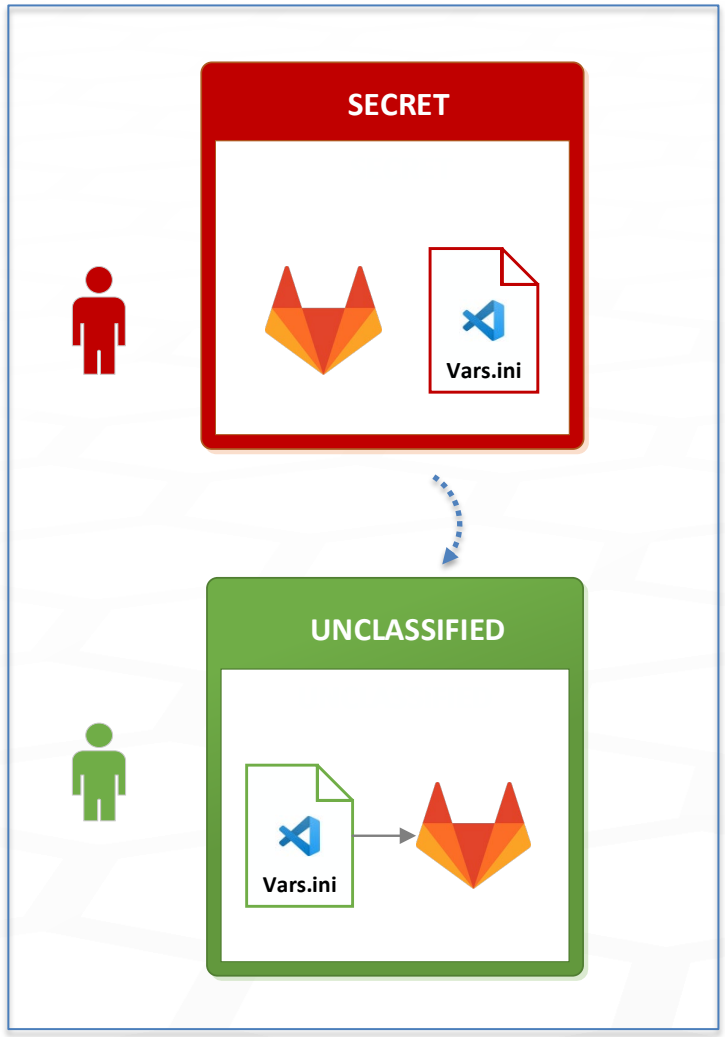
# Pattern 3: Use Multilevel Share to Collaborate and Share DE Labelled Models

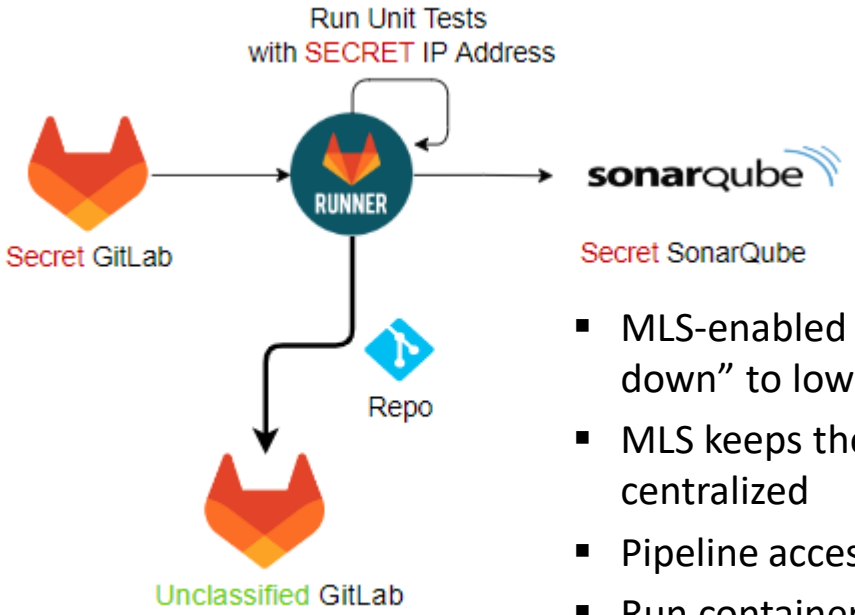
1. Polyinstantiate MagicDraw
   - Authors update models at multiple levels from single pane of glass
2. Containerize and Deploy TWC at Multiple Levels
4. Enforce separation at the network level [ZT]
6. Consolidate license server into MLS environment for efficient license management




3. Deploy multilevel file share to store labelled data
5. Limit access to resource based on sensitivity of the resource and authorization of the user [ZT]

# Pattern 4: Consolidate Classified CI/CD Pipelines with MLS




- MLS-enabled GitLab Runners automatically “read down” to lower level repositories for data ingest
- MLS keeps the pipelines isolated, segregated, and centralized
- Pipeline access controlled by MLS security policy
- Run containerized software and execute within the CI/CD Pipeline

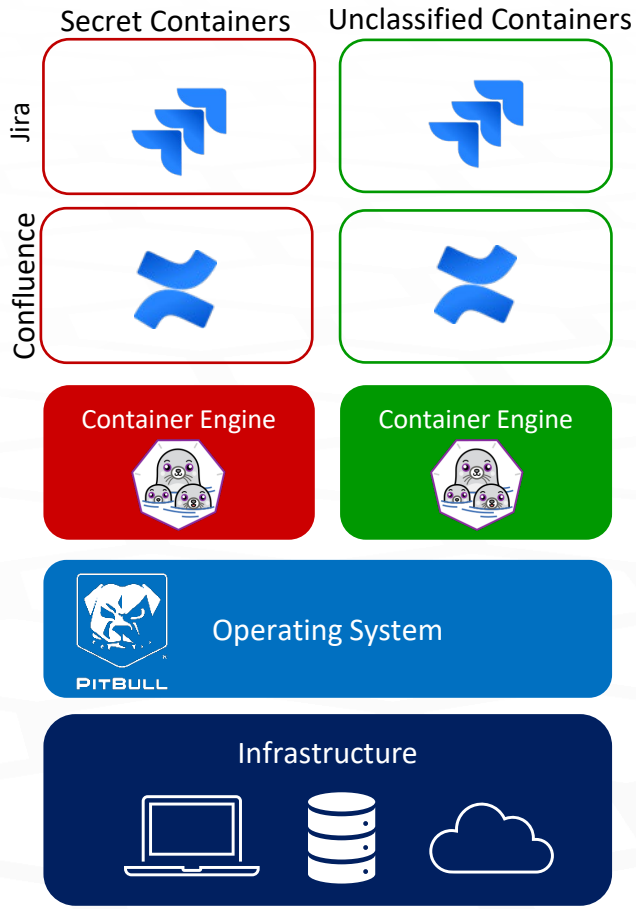
- Deploy to the cloud, local virtualized environments, or physical hardware while maintaining existing security profiles

Developers write code at various levels that is seamlessly pulled into a single pipeline, improving collaboration and data integrity

# Pattern 5: Jira and Confluence in Multilevel Secure Containers

- System administrators prepopulate the container registry with Jira and Confluence containers to deploy on-demand each application at the desired sensitivity level
- Data elevator is used to move Jira or Confluence data from one sensitivity level instance to another sensitivity level instance
- Host the data elevator in a GitLab runner to auto-move the data between sensitivity levels at a predetermined interval within a single MLS environment

Data is labeled and segregated on the back end while users log into a single tool and can see all their data in one screen



# Digital and Development Engineering Tools Coverage

| Tool has been deployed using … | Containers | Polyinstantiation | By |
|---|---|---|---|
| Cameo Systems Modeler | | ✓ | 🤝 |
| | | ✓ | truMLS |
| BigLever Software, Inc. | | ✓ | truMLS |
| Teamwork Cloud | ✓ | | truMLS 🤝 |
| | ✓ | | truMLS |
| | ✓ | | truMLS |

| Tool has been deployed using … | Containers | By |
|---|---|---|
| | ✓ | truMLS 🤝 |
| Confluence | ✓ | truMLS 🤝 |
| JIRA | ✓ | truMLS 🤝 |
| Mattermost | ✓ | 🤝 |
| sonarqube | ✓ | truMLS 🤝 |
| | ✓ | truMLS |

**Future focus areas for tool interoperability:**

windchill®

All third party tools listed have been run in a multilevel environment with no changes made to the underlying commercial tool.

New tools are being integrated every day following the same patterns shown here. More to come soon!

# Containerized Digital Engineering Applications through Multilevel Secure Architectures

- Containerize commercial digital engineering applications for multilevel environment
- Polyinstantiate applications for usability (single pane of glass)
- Facilitate data centric security model on premise and in the cloud
- Reduce infrastructure, both physical and virtual
- Enable Zero Trust based MLS capabilities through commercial trusted operating system

## For More Information

Connect with Scott: scott.devitt@gd-ms.com | 952-921-6411

Visit our Website: https://gdmissionsystems.com/products/multilevel-security/trumls

truMLS®