Building Secure Cross-Domain Solutions with Kasm and Teleport

March 27, 2026 / Gavin Jackson

kasm teleport security remote-desktop zero-trust cross-domain infrastructure

Cross-domain solutions are hard.

Not technically impossible - we have the tools. But the combination of strict security requirements, usability concerns, and audit compliance makes them one of the more challenging infrastructure problems to solve well.

I have been looking at this problem from a few angles recently. The goal: allow users to access resources in different security domains without creating pathways for data leakage or unauthorised access between those domains.

Two tools have emerged as particularly useful in this space: **Kasm Workspaces** for isolated, ephemeral browser sessions, and **Teleport** for zero-trust access control and session recording.

Here is how they fit together.

The Problem

In many organisations - especially government, defence, finance, and healthcare - information is classified into different security domains. A user might need to:

- Browse the public internet (low trust)
- Access internal corporate systems (medium trust)
- Work with classified or sensitive data (high trust)

Traditionally, this meant air-gapped machines, multiple physical workstations, or heavily locked-down VDI environments that users hate.

The modern approach is to use **isolated, ephemeral workspaces** that can be spun up on demand, used for a specific task, and then destroyed - leaving no persistent data on the endpoint.

Kasm Workspaces: The Isolation Layer

Kasm Workspaces is an open-source (with commercial support) container streaming platform. It delivers browser-based access to isolated applications and desktops running in Docker containers.

Why Kasm Works for Cross-Domain

Ephemeral by design: Every session starts fresh from a clean image. When the user closes the browser tab, the container is destroyed. No persistence, no malware surviving reboots, no data left behind.

Browser-based access: Users do not install anything. They open a URL, authenticate, and get a fully functional Linux desktop or single application streaming to their browser via WebRTC.

Granular images: You can define different images for different trust levels:

- A hardened Chrome image for general web browsing
- A Firefox image with specific extensions for research
- A full Ubuntu desktop for development work
- A locked-down image with no clipboard, no downloads, no printing for highly sensitive access

Network segmentation: Each Kasm agent can be placed in a different network segment. The workspace container only has access to the networks you explicitly allow.

Setting Up Domain Isolation

The key architectural decision is mapping Kasm workspaces to security domains:

```
+-------------------------------------------------------------+
| Kasm Manager |
| (Orchestration and User Auth) |
+-------------------------------------------------------------+
 |
 +--------------------+--------------------+
 v v v
+---------------+ +---------------+ +---------------+
| Kasm Agent | | Kasm Agent | | Kasm Agent |
| (Domain A) | | (Domain B) | | (Domain C) |
| Internet | | Corporate | | Classified |
| Zone | | Network | | Network |
+---------------+ +---------------+ +---------------+
```

Each Kasm agent runs on a host with specific network connectivity. The "Internet Zone" agent has outbound internet access. The "Classified Network" agent only has access to classified resources. The user chooses (or is assigned) the appropriate workspace based on their current task.

Kasm Configuration Tips

Disable persistence features for high-trust domains:

```
# In your Kasm image configuration
persistent_profile: false
enable_clipboard: false
enable_downloads: false
enable_uploads: false
```

Use network policies to restrict container egress:

```
# Docker daemon configuration or Kubernetes network policies
egress:
 - to:
 - namespaceSelector:
 matchLabels:
 domain: classified-resources
```

Enable session recording for audit trails:

Kasm can record sessions as video files, which is useful for compliance. But for a more integrated approach, we bring in Teleport.

Teleport: The Access Control Layer

Teleport is an open-source (with enterprise features) access platform that provides:

- **Certificate-based authentication:** No static credentials, short-lived certificates
- **Role-based access control (RBAC):** Fine-grained permissions based on identity
- **Session recording:** Complete audit trail of every command, query, or desktop session
- **Just-in-time access:** Request and approve access workflows
- **Unified access:** One tool for SSH, Kubernetes, databases, applications, and desktops

Why Teleport Complements Kasm

While Kasm provides isolation, Teleport provides **identity, audit, and policy enforcement**.

Unified identity: Teleport can act as the identity provider for Kasm, or integrate with your existing IdP (Okta, Azure AD, etc.) and pass identity attributes through to Kasm.

Policy enforcement: Teleport's RBAC can determine which users can access which Kasm workspaces based on labels, time of day, and approval workflows.

Enhanced audit: Teleport records not just that a user accessed Kasm, but what they did inside the session - with structured logs that integrate with SIEM tools.

Secure bastion: Teleport can act as the only entry point to your Kasm infrastructure, eliminating exposed Kasm manager interfaces.

Integrating Kasm with Teleport

The cleanest way to combine the two is to make Teleport the front door and policy layer for every Kasm environment. Users authenticate to Teleport first, then launch the Kasm workspace that matches the domain they are allowed to access.

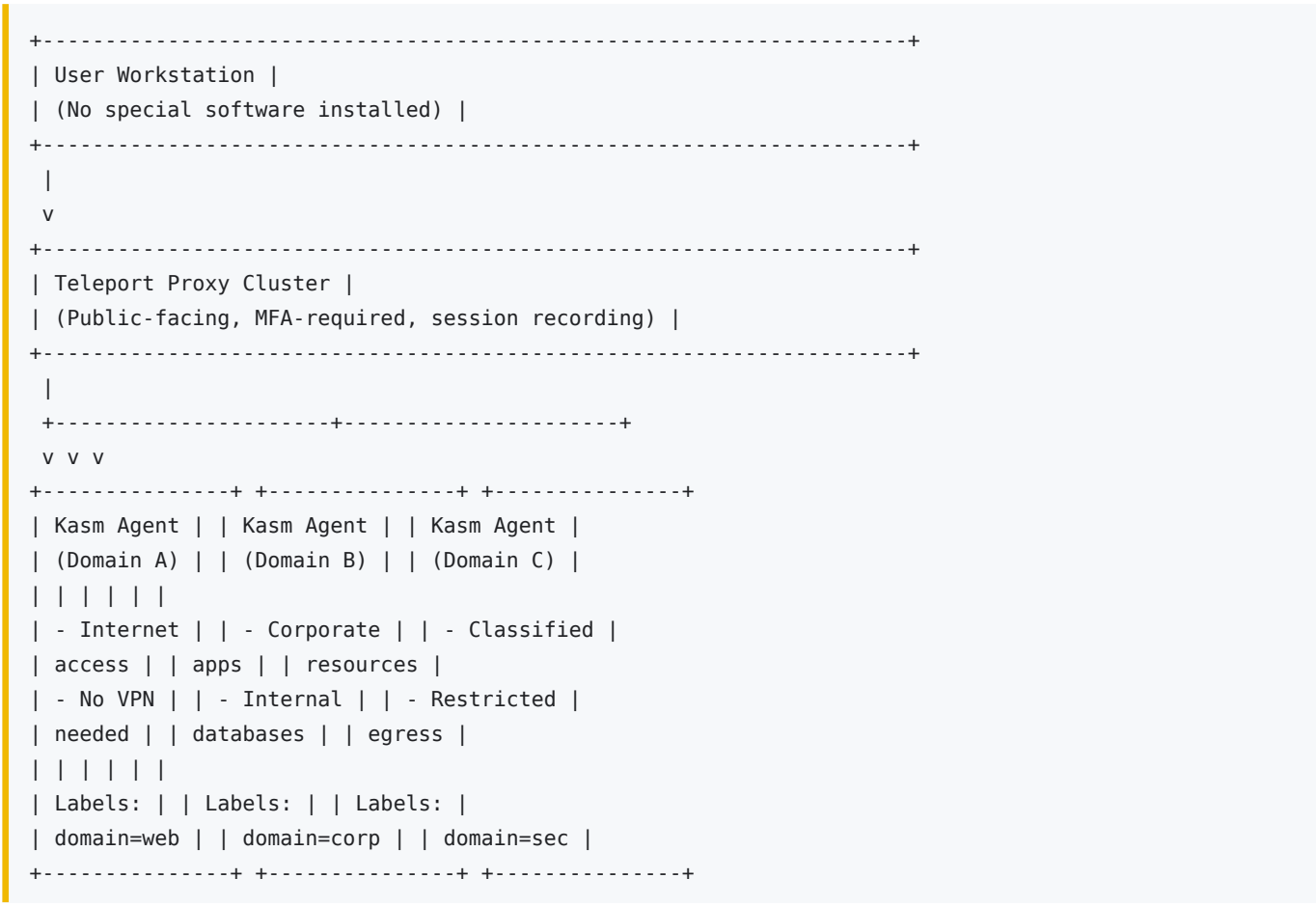
That keeps the access path consistent:

```
User -> Teleport Proxy -> Kasm Manager -> Kasm Agent (Domain X)
```

Teleport decides which Kasm applications appear to the user, enforces MFA and session limits, and records the access event. Kasm then provides the isolated workspace inside the correct network segment.

Putting It Together: A Reference Architecture

Here is what that deployment looks like in practice for a three-domain cross-domain solution:



In this model, each Kasm manager endpoint is published into Teleport as an application with labels that represent the relevant domain and classification. Users never browse directly to Kasm. They go through Teleport, which becomes the single access path for internet, corporate, and classified workspaces.

Teleport Application Mapping

```
# teleport.yaml app configuration
app_service:
 enabled: true
 apps:
 - name: kasm-internet
 uri: https://kasm-manager.domain-a.internal
 labels:
 domain: web
 classification: unclassified

 - name: kasm-corporate
 uri: https://kasm-manager.domain-b.internal
 labels:
 domain: corp
 classification: internal

 - name: kasm-classified
 uri: https://kasm-manager.domain-c.internal
 labels:
 domain: sec
 classification: secret
```

From the user side, the experience stays simple:

```
# List available Kasm environments
tsh apps ls

# Launch the classified Kasm environment
tsh app launch kasm-classified
```

Teleport RBAC Mapping

```
# Role: general-user
allow:
 app_labels:
 domain: ["web", "corp"]

# Role: cleared-researcher
allow:
 app_labels:
 domain: ["web", "corp", "sec"]
 require_session_mfa: hardware
 request_access: optional

# Role: admin
deny:
 app_labels:
 domain: ["sec"] # Admins don't automatically get classified access
```

Data Flow Controls

To prevent data exfiltration between domains:

1. **Kasm-level:** Disable clipboard, file upload/download, printing for high-trust domains
2. **Network-level:** Strict egress filtering from Kasm agents - only allow connections to known resources
3. **Teleport-level:** Session recording, identity-aware access policies, and detailed audit logs
4. **Process-level:** DLP scanning on any approved export pathways (if required)

Operational Considerations

Performance

Kasm streams desktop environments via WebRTC. For acceptable performance:

- Kasm agents should be geographically close to users
- Allocate sufficient CPU/memory for concurrent sessions
- Use GPU acceleration for video-heavy workloads

Teleport adds minimal latency for application access but can proxy desktop protocols if needed.

Scaling

- **Kasm:** Scale agents horizontally based on concurrent session demand
- **Teleport:** Proxy cluster can scale behind a load balancer; auth service is the bottleneck (usually fine for thousands of users)

Backup and Disaster Recovery

- Kasm workspaces are ephemeral - no backup needed
- Kasm configuration (images, settings) should be in Git/IaC
- Teleport cluster state should be backed up (etcd or DynamoDB depending on deployment)

Monitoring

Key metrics to track:

- Concurrent Kasm sessions per domain
- Session duration and idle time
- Teleport authentication failures
- Resource access patterns (unusual database queries, etc.)

Conclusion

Cross-domain solutions do not have to mean clunky VDI or multiple physical machines. With Kasm Workspaces for isolation and Teleport for access control, you can build a modern, user-friendly system that satisfies security requirements without making users miserable.

This approach gives you:

- **Ephemeral, isolated workspaces** per security domain
- **Zero-trust access** with short-lived certificates
- **Complete audit trails** for compliance
- **Browser-based access** - no client software to manage
- **Granular controls** over data flow between domains

It is not a turnkey solution - you still need to design your network segmentation, define your RBAC policies, and train your users. But the tools are there, they are open source (with commercial support options), and they work.

If you are wrestling with cross-domain access problems, this stack is worth evaluating.

Resources:

- Kasm Workspaces
- Teleport
- Kasm Documentation - Workspaces
- Teleport Documentation - Architecture
