

Title - Drupal-Powered Client-Server DAM for Multi-Site Media

Introduction

What if you could manage and share media across dozens of Drupal sites from a single dashboard? Discover a lightweight, JSON:API-powered DAM solution that scales with your ecosystem in minutes.

Prerequisite

To get the most out of this session, you should already know:

- Drupal 10/11 site-building and basic custom module development.
- How Media entities work (Media source plugins, fields).
- JSON:API and REST API fundamentals in Drupal.
- Role-based permissions and authentication concepts (JWT or Basic authentication).

Outline

1. Why a Centralized DAM?
 - a. We'll begin by outlining the drawbacks of siloed media—wasted storage, version drift, and editorial friction—and demonstrate how a single “parent” DAM can serve “consumer” sites with on-demand, consistent media delivery.
2. drupal_dam Module (Parent Site)
 - a. Enabling JSON:API on Field Storage & Field Config
 - b. Creating a dam_consumer role & assigning precise API permissions
 - c. Building a form to select which media bundles to expose
 - d. Exposing media via a secure custom REST endpoint
3. drupal_dam_client Module (Child Sites)
 - a. Configuration form: connecting to parent JSON:API & entering JWT credentials
 - b. Enhancing the Media > Add page with “Upload to Parent DAM” links
 - c. Dynamic form generation on media-create:
4. Fetching field definitions via JSON:API
 - a. We'll cover JSON:API calls to pull help text, required flags, defaults, and option values—then auto-build entity reference and list fields. As a bonus, we'll discuss pre-populating edit forms via the same mechanism.
5. Media Library Integration

- a. See how to add a “Parent DAM” tab to each exposed bundle, enabling in-UI browsing and one-click media selection from the centralized repository
6. Lessons Learned & Future Enhancements
 - a. Performance considerations, caching strategies
 - b. Multi-site scalability tips
 - c. AI-powered enhancements: automatic duplicate detection and removal using AI image/text analysis, and semantic search on the Parent DAM to improve asset discovery

Learning Objectives

By the end of this session, you will be able to:

- Architect a parent/consumer DAM topology in Drupal
- Securely expose selected media bundles via a custom REST endpoint
- Configure and consume remote media in child sites using JSON:API and JWT
- Dynamically generate media-upload forms based on remote field definitions
- Integrate external media sources directly into the Media Library UI
- Anticipate and solve common scaling and performance challenges