Adobe/SHRSS KT Session: Tagging, Taxonomy & Metadata Governance

Authoritative references used:

- Taxonomy and tagging best practices for AEM Assets
- Tags, taxonomy, and metadata best practices: high-level summary (Site hierarchy)
- `Content package: /SHRSS_Knowledge_Transfer/Content/shrss-content-minimal-assets-PROD-1.0`

---

1. Tag Taxonomy

1.1 How taxonomy was configured for this implementation

- **Taxonomy root:** Tags are stored under the standard AEM taxonomy root: `/content/cq:tags/shrss`, under the **shrss** namespace for SHRSS-specific tags.
- **Path-to-tag mapping (site/content context):**
  - The implementation uses the custom **TagsPathMappingConfigService**, which reads a path-to-tag mapping from an **ACS Commons Generic List** at `/etc/acs-commons/lists/path-tags-mapping/jcr:content/list`.
    - Each list item has:
      - **jcr:title** = content path (e.g. site or section path)
      - **value** = tag ID (e.g. a tag under the taxonomy)
  - This allows the system to **derive tags from the current page path** (e.g. for alerts or context-specific content) without authors selecting tags manually on every page.
- **Content Fragments and assets:**
  - Content Fragments and DAM assets store tag references in metadata:
    - For CFs, the **categories** field (multi-value) holds tag IDs used for filtering in dynamic lists.
    - DAM assets use the standard **cq:tags** property on `jcr:content/metadata` (with `cq:Taggable` mixin).
      - Examples:
        - `shrss:regions/latam`
        - `shrss:event-categories/entertainment`.
- **Tag namespaces in use (from code and test data):** `shrss` (e.g. `shrss:regions/apac`, `shrss:event-categories/entertainment`, `shrss:hotel/new-york`), and standard OOTB namespaces such as `properties` (e.g. `properties:orientation/portrait`).
  - Spreadsheet:

/SHRSS_Knowledge_Transfer/Tagging_and_Taxonomy/SHRSS_Content_Tag_Mapping.xlsx`

1.2 How tags are structured across Brand / LOB / Property

- **Namespace as top level:** Under `/content/cq:tags/shrss`, tags can be organized by brand, LOB, or property using **child tag hierarchies** (e.g. regions, event-categories, hotel/property names).
- **Observed structure (from implementation and test data):**
    - **Region/destination:** e.g. `shrss:regions/latam`, `shrss:regions/apac`
    - **Event/content type:** e.g. `shrss:event-categories/entertainment`
    - **Property/venue:** e.g. `shrss:hotel/new-york`
- **Path–tag mapping:** The ACS Commons list maps **site/section paths** to **tags**, so a given brand or property section can be associated with one or more tags for automatic context (e.g. which alerts or content to show). This supports "tags by brand/LOB/property" without duplicating taxonomy; the same tag taxonomy is reused and mapped per path.

1.3 Best practices and taxonomy strategy

- **Sites (hierarchy and governance):** Use folder structure and permissions to control who can access which areas; use tags for **structural metadata** (categorization, related content, navigation). Prefer a **defined taxonomy** over ad-hoc tags to keep search and filtering predictable. See Site hierarchy, taxonomy, and tagging guide.
- **Assets:** Use a **hierarchical taxonomy** (namespaces → tags → sub-tags) so authors can select from consistent keywords; this improves findability and faceted search. Make taxonomy **additive to** (not duplicative of) folder structure and metadata. Consider **Smart Tags** to reduce manual tagging. Govern who can create/edit tags via permissions. See Taxonomy and tagging best practices for AEM Assets.
- **SHRSS-specific:** Keep path–tag mapping in the ACS Commons list documented and maintained when site structure or brands change. Use the **shrss** namespace for all SHRSS-specific taxonomy; use standard namespaces (e.g. **properties**) only where they add value (e.g. orientation, format).

2. Tag Management

2.1 Governance — Rules for tag governance (who creates, who maintains, who approves)

- **Recommended model (align to your organization):**
    - **Create:** Limit tag creation to a small group (e.g. admins or a "taxonomy steward") so the hierarchy stays consistent.
    - **Maintain:** Designate owners per namespace or domain (e.g. regions, event-categories); they review and prune unused or duplicate tags.
    - **Approve:** For sensitive or brand-critical namespaces, require approval before new tags go live

(process outside AEM or via workflow if needed).

- **AEM mechanics:** Use **permissions on the tag tree** (`/content/cq:tags`) to restrict create/update/delete by group. Restrict the **path–tag mapping list** (`/etc/acs-commons/lists/path-tags-mapping`) to admins so only authorized users change which paths map to which tags.
- **Documentation:** Keep a short governance doc: who can create/edit tags, who maintains the path–tag list, how to request new tags or mappings, and where the taxonomy is documented (e.g. Tagging console, spreadsheet, or wiki).

2.2 Creating, updating, deleting tags

- **Creating:** Tagging console → select parent namespace or tag → Create Tag. Provide **Title** (and optional **Name**, **Description**). For SHRSS, create under `/content/cq:tags/shrss` or the appropriate sub-hierarchy.
- **Updating:** Tagging console → select tag → Edit. Change Title/Name/Description. Moving tags (Move) can break references; plan and communicate.
- **Deleting:** Tagging console → select tag → Delete. Container tags may require children to be moved or deleted first. **References** should be reviewed before delete so content is retagged if needed.
- **Publishing:** Tags must be **published** to be available on publish; use Publish/Unpublish from the Tagging console. Path–tag mapping content (ACS Commons list) must also be published if author and publish use it.

2.3 Best practices

- **Standardize:** Use a single vocabulary/glossary
- **Avoid synonyms and near-duplicates:** Avoid synonyms as separate tags unless using thesaurus features.(e.g., don’t have both `footwear` and `shoes` for the same concept).
- **Create namespaces:** for major domains: `brand`, `product`, `experience`, `audience`, etc.
- **Keep hierarchies shallow but expressive:** – more like 3–5 levels, not 10.
- **Avoid over-tagging:** Too many tags per asset/page dilutes value; define guidelines (e.g. max tags per content type).
- **Re-evaluate:** Periodically audit tag usage and merge or retire unused tags.
- **Control who can create tags:** no ad-hoc tags from every author.
- **Periodically review and refactor:** merge, deprecate, localize.
- **Path–tag list:** Treat the ACS Commons path–tag list as governed configuration; change only with approval and document each path–tag pair for troubleshooting.

---

3. Application of Tags in SHRSS Authoring & Asset Management

3.1 Component and asset metadata mapping

- **Assets (DAM):** Metadata schema is defined at `conf/global/settings/dam/adminui-extension/metadataschema/shrssmetadataschema`. The **Tags** field is typically bound to `jcr:content/metadata/cq:tags`. Authors pick tags from the taxonomy when editing asset metadata; tags are stored as string array on the asset.
- **Content Fragments:** CF models (e.g. News, Events) include a **categories** field that stores tag IDs. This is used by **CF Card List** and related components to filter by tags. Map this field in the CF model to the same taxonomy used in the Tagging console.
- **Pages:** Page properties can expose **Tags/Keywords** (cq:tags) for SEO and for components that use page context (e.g. Alert Aggregator uses path–tag mapping derived from current page path).
- **Path–tag mapping:** Not a component field; it is authoring-time configuration in the ACS Commons list. It maps **paths** (e.g. `/content/shrss/hotel/en/...`) to **tag IDs** used by backend services (e.g. **TagsPathMappingConfigService.getAllTagsByPath(currentPage.getPath())**).

3.2 How tags impact dynamic content queries

- **CF Card List (listType = tags):** When authors choose **List by tags**, they configure **tagsList** (tag IDs) and **tagsRootFolder** (e.g. `/content/dam/shrss`). The backend builds a Query Builder map that:
  - Restricts to DAM assets that are Content Fragments under the root folder.
  - Filters by **content fragment model** (e.g. Event, News).
  - Adds **group property** conditions on the **categories** property matching each tag in **tagsList** (OR logic between tags).
    So only CFs whose **categories** metadata contains at least one of the selected tags appear in the list.
  - `shrss/news-categories/featured-news`
    - https://author-p135156-e1336227.adobeaemcloud.com/ui#/aem/editor.html/content/shrss/corporate/hardrock/en/blog.html
- **Path-derived tags: Alert Aggregator** uses service to get tags for the current page path, then uses those tags to decide which alerts to show. So **path–tag mapping** directly drives which content appears on a given page/section.
  - *TagsPathMappingConfigService.getAllTagsByPath(currentPage.getPath())*

3.3 Category filters

- **CF Card List:** The same **tagsList** used for the list query can be exposed as category filters in the UI (e.g. news search result filters by category/tag). Tag titles are resolved via **TagUtils.getTagNameTitleMapByTagNames** / **TagManager** for display.
- **Asset search:** In Assets, the Tags predicate in search (e.g. Search Rail) can be configured with a **root tags path** (e.g. `/content/cq:tags/shrss`) so authors filter by the SHRSS taxonomy.

3.4 Search results

- **Assets:** Tags are indexed with asset metadata; search on `jcr:content/metadata/cq:tags` (or equivalent) returns assets that have the given tag(s). Match-all vs match-any is configurable in the Tags predicate.
- **Content Fragments:** The **categories** field is part of CF metadata and is queryable; tag-based CF lists and search use this for filtering.
- **Pages:** If page properties store cq:tags, those can be used in page search and for SEO (meta keywords).

3.5 Content inheritance across sites

- **No JCR inheritance of tags:** Tags are not inherited from parent pages by default; each page/asset holds its own tag references.
- **Path–tag mapping as “inheritance”:** The path–tag list effectively encodes “sections of the site and their tag context.” When a page’s path is under a mapped path, **getAllTagsByPath** returns the tags for that path (and any parent path that has a mapping). So “inheritance” of tag context is via **path prefix matching** in the mapping list, not via JCR hierarchy.

3.6 Card and page visibility

- **Cards (CF Card List):** Visibility is driven by **tagsList** and **tagsRootFolder** plus CF model type. Only CFs with matching **categories** (tags) under the root folder appear. Wrong or missing tags on CFs, or wrong tagsList/tagsRootFolder on the component, will hide cards.
- **Page visibility:** For components that use **path–tag mapping** (e.g. Alert Aggregator), visibility depends on the current page path having a mapping and the mapped tags matching the content (e.g. alert tags). Missing or incorrect path–tag entries will cause content not to appear on that section.

4. Additional Topics (brief)

- **Tag performance (query efficiency, indexing):** Tag properties are indexed when standard AEM/Oak indexing is used. Queries that filter by `cq:tags` or CF **categories** should use indexed properties; avoid large OR sets of tags in a single query when possible. Path–tag mapping is a small list read once per request; keep the list size reasonable.
- **Troubleshooting content not appearing in listings/filters:** (1) Confirm the CF or asset has the expected tags in **categories** or **cq:tags**. (2) Confirm component config: **tagsList**, **tagsRootFolder**, and list type (e.g. tags vs fixed list vs root path). (3) For path-driven content (e.g. alerts), confirm path–tag mapping exists for the page path and is published. (4) Confirm tags are published. (5) Check Query Builder or service logs if queries return no results.
- **Scheduling or lifecycle states:** Tags themselves do not change with scheduling; they are static metadata. If “visibility” is time-based, that is usually implemented via **publish date / end date** on the CF or page, or via workflow state, not via tag changes. Tags can still be used **together** with date filters (e.g. Event CF list filtered by tag and by event date range).

Appendix A: Asset Metadata Fields Configured for Tag Selection

Source: `/conf` in content package `Content/shrss-content-minimal-assets-PROD-1.0/jcr_root/conf`

All metadata fields below use a **tagfield** dialog field element (values selected from the tag taxonomy). Widget: `cq/gui/components/coral/common/form/tagfield` with `metaType="tags"`.

1. DAM Asset Metadata Schema (Assets UI)

Configuration path: `/conf/global/settings/dam/adminui-extension/metadataschema/shrssmetadataschema`

Applied to assets in the Assets console (metadata form). Stored under `jcr:content/metadata/`.

Field label	Metadata property	Tag root path (selection scope)
Tags	`cq:tags`	*(not set – full taxonomy)*
Brand	`shrssbrands`	`/content/cq:tags/shrss/brands`
Venues & Branded Experiences	`shrssvenues`	`/content/cq:tags/shrss/venues-and-branded-experiences`
Line of Business	`shrsslob`	`/content/cq:tags/shrss/lob`
Property Names (Locators / City Drop)	`shrssproperty`	`/content/cq:tags/shrss/properties`
Type	`shrsstype`	`/content/cq:tags/shrss/type`
Category	`shrsscategory`	`/content/cq:tags/shrss/category`
Product	`shrssproduct`	`/content/cq:tags/shrss/products`

2. Content Fragment Models (CF authoring)

Configuration path: `/conf/shrss/settings/dam/cfm/models/<model>`

Used when editing Content Fragments. Values stored in the CF model data (e.g. `jcr:content/data/master/`).

Events model

`conf/shrss/settings/dam/cfm/models/events`

Field label	Property name	Tag root path
Categories	`categories`	`/content/cq:tags/shrss/event-categories`

News model

`conf/shrss/settings/dam/cfm/models/news`

Field label	Property name	Tag root path
Categories	`categories`	`/content/cq:tags/shrss`
Tags	`tags`	`/content/cq:tags`

Summary list (all metadata fields that use tag selection)

1. **DAM metadata (shrssmetadataschema):** Tags (`cq:tags`), Brand (`shrssbrands`), Venues & Branded Experiences (`shrssvenues`), Line of Business (`shrsslob`), Property Names (`shrssproperty`), Type (`shrsstype`), Category (`shrsscategory`), Product (`shrssproduct`).
2. **Content Fragment – Events:** Categories (`categories`) — root: `shrss/event-categories`.
3. **Content Fragment – News:** Categories (`categories`) — root: `shrss`; Tags (`tags`) — root: full taxonomy.

Note: “Smart Color Tags” and “Smart Tags” (predictedTags) are tag-related but not manual tag selection; they are system/AI-driven and are not included in this list.

Appendix B: Tagging versus Asset Metadata

Overview

- **Metadata** = structured properties *about* an asset (title, campaign, usage rights, SKU, dates…). In AEM Assets these are defined and surfaced via **metadata schemas** and profiles, stored under `jcr:content/metadata` on the asset node.
- **Tags** = reusable **taxonomy terms** (`cq:Tag`) applied as metadata (usually `cq:tags`) to categorize content by business concepts (region, product, audience, theme). They create flexible, hierarchical classification across Sites and Assets.
- **When to use what**
  - Use **metadata fields** for *fixed, well-defined attributes* that behave like columns in a table (single source of truth, integrations, governance, permissions).
  - Use **tags** for *business taxonomy* and *faceted discovery* (findability, related content, navigation, personalization).
- Authors see both in the same properties dialog: **fields** (text, dropdowns, dates…) vs. **tag pickers**. Technically both end up as metadata; the difference is whether the value comes from a **schema field** or from the **central tag taxonomy**.

What is asset metadata in AEM?

From AEM's perspective, asset metadata is:

- Data stored primarily under `assetPath/jcr:content/metadata` on a `dam:Asset` node, using standard namespaces like `dc:*`, `xmp:*`, `tiff:*`, plus custom properties.
- Used to answer questions like:
  - **What is this?** (title, description, format)
  - **Who is it for / about?** (audience, brand, market, persona)
  - **Where and how can it be used?** (license, channels, regions)
  - **When is it valid?** (valid from/to, campaign dates)

Metadata is the **backbone of structured content** in Assets – essential for search, automation, personalization and governance "Metadata best practices for AEM Assets" and "Metadata overview" confirm this.

What is tagging in AEM?

> ⓘ **Note**
>
> The official "Tags, Taxonomy, and Metadata Best Practices" guide calls tags **pre-defined terms** that normalize how content is described and power faceted search, related content, SEO, etc. "Site Hierarchy, Taxonomy, and Tagging Guide".

Tags in AEM are:

- Nodes of type `cq:Tag` stored in the tag taxonomy (namespaces + hierarchies).
- Applied to assets and pages via properties like `cq:tags` and used for organizing, searching, navigation, personalization, etc. See **Using Tags** and **Administering Tags** for AEMaaCS Sites and Assets "Using Tags", "Administering Tags".
- Architecturally a **global, reusable vocabulary** independent of any single asset or page. The taxonomy can be centrally governed and reused across:
  - Assets (DAM)
  - Sites pages and Experience Fragments
  - Content Fragments/headless content
  - Content Hub / external consumers via `xcm:keywords` mappings "Taxonomy management in Assets view".

Relationship: tags *are* metadata, but not all metadata are tags

Architecturally:

- Tags are *implemented as metadata* (usually `cq:tags` string array), and AEM's search/indexing treats them as such.
- But conceptually, tags are the **controlled vocabulary layer** on top of raw fields – a reusable semantic

layer across many assets/pages.

- Metadata schemas define *where* tags are stored and *how* they are presented to authors (Tag field vs. text, dropdown, etc.) “Metadata schemas”.

Choose a *metadata field* when…

Use a schema field (text, dropdown, date, boolean, etc.) when the attribute is:

1. **Object-like and single-source-of-truth**
   - IDs: SKU, product code, asset ID, campaign ID.
   - Legal / regulatory: license ID, contract number.
2. **Strongly typed or constrained**
   - Dates (start/end), numeric values, booleans, fixed enums.
   - Data that is validated or used in external systems / integrations.
3. **Operational or system-internal**
   - Workflow state, approval status, archival flag, retention classification.
   - Permissions drivers (metadata-driven permissions in AEMaaCS) “Metadata-driven permissions”.
4. **Not meant for faceted navigation across many content types**
   - For example, an internal “Upload Batch ID” or “Ingestion Source” that rarely drives end-user filters.
5. **Needs to be different per schema / per folder**
   - Specific metadata only applicable to product assets vs. brand imagery, etc.

Examples where metadata fields are better than tags

Use case	Recommended implementation
Product SKU / variant code	Custom text field `commerce:sku` in schema
License expiration date	Date field `xmpRights:ExpirationDate`
Rights territory	Controlled dropdown, possibly multi-select
Campaign ID / Salesforce ID	Text field, validated, used in downstream systems
Internal lifecycle flags	Boolean or enum fields (Approved, Archive, etc.)

Choose *tags* when…

Use tags when the attribute is:

1. **Taxonomic / conceptual**
   - Product categories, collections, themes, genres, topics.
   - Audiences (musicians, families, loyalty tiers).

    - Locations as conceptual facets (e.g. “Latin America” vs. precise geo-code).

2. **Shared across many content types and systems**
    - You want one global vocabulary for assets, pages, CFs, Content Hub, etc.
    - You plan to reuse the same facets for navigation, recommendations, personalization.
3. **Many-to-many and evolving**
    - An asset can belong to multiple campaigns, audiences, themes.
    - Business wants to extend/refine the taxonomy without touching code/schemas.
4. **Driving discovery, not transactionality**
    - Faceted search filters (“Region”, “Brand”, “Season”).
    - Related content, “More like this”, “browse by theme”.
5. **Useful for non-technical governance**
    - Taxonomy and tag governance can be managed by business owners, with roles and permissions, without schema changes “Taxonomy and tagging best practices for AEM Assets”.

Examples where tags are better than metadata fields

Use case	Recommended implementation
Brand / Sub-brand facets	`brand:hard-rock/cafe`, `brand:hard-rock/hotel` tags
Experience type (dining, gaming, live)	`experience:dining`, `experience:gaming`, `experience:live`
Campaign / initiative themes	`campaign:summer-2026`, `campaign:grammys`
Audience segments	`audience:families`, `audience:vip`, `audience:locals`
Content topics	`topic:music/rock`, `topic:food/burgers`

Common combination patterns

You often use **both**:

- **Product images**
    - Metadata: `commerce:sku`, `commerce:color`, `dc:format`, license dates.
    - Tags: product category, collection, season, region.
- **Campaign hero images**
    - Metadata: campaign code, start/end date, legal disclaimers.
    - Tags: campaign name, audience, channel, theme, region.
- **Restaurant / hotel photography**
    - Metadata: location code, photographer, usage rights.

- Tags: property, venue type (pool, bar, lobby), time of day, mood.