TERMS OF ART

A CONTENT MANAGEMENT CARD GAME

45 terms · 12 doctrines · 12 scenarios · 3 to 6 players

This deck is not flashcards. Flashcards test whether you can recall a definition; they do not test whether you can hold a boundary under pressure. The vocabulary of content management is unusually treacherous — the same word means different things in different systems, and several pairs of terms are, by the source glossary's own admission, routinely confused and used interchangeably.

So this game makes students argue. Every round ends with somebody defending a usage out loud while everyone else looks for the hole in it. The definitions get read aloud, in context, repeatedly, because that is how the scoring works — not because a card told someone to flip it over.

WHAT IS IN THE BOX

- **45 Term cards —** Five suits of nine. Each carries the term, the other names it goes by, and the term it is most often confused with.
- **12 Doctrine cards —** Principles of content management. These set the subject of each round and are the scoring tokens.
- **12 Scenario strips —** Concrete situations. Used in the third variant.
- **The Key —** Full definitions for all 45 terms, plus a confusion table. The Arbiter holds this and rules with it.

The one rule that makes this work

A player may always challenge another player's usage. The Key settles it. Being right is worth points; bluffing is not free.

BEFORE YOU PLAY

How to print and assemble

Everything is laid out nine-up on US Letter at true poker size (2.5 × 3.5 in). Card backs are identical within each deck, so no mirroring is required — a straight duplex job lines up correctly.

- **Pages 6–15 — Term deck.** Print double-sided, flip on LONG edge. Five sheets, 45 cards.
- **Pages 16–19 — Doctrine deck.** Print double-sided, flip on long edge. Two sheets, 12 cards.
- **Page 5 — Scenarios.** Print single-sided. Cut into 12 strips.
- **Pages 20–26 — The Key.** Print single-sided. One copy is enough; the Arbiter holds it. Page 26 is the confusion table used in Variant One.

Set the printer to 100% / Actual Size, never Fit to Page, or the card grid will drift and your cuts will wander. Card stock between 200 and 250 gsm behaves best; ordinary paper works if you sleeve the cards. Cut lines are printed as hairlines around every card, with registration ticks in the page margins for anyone using a guillotine.

IF YOU ONLY HAVE ONE-SIDED PRINTING

Print the fronts only and skip the back pages. The game plays fine with blank backs in every mode except the Doctrine draft, where players benefit from not seeing what others hold. In that case, sleeve the cards with any opaque backing card behind them.

THE FIVE SUITS

STRUCTURE	How content is shaped	STR-1 to STR-9
LIFECYCLE	How content moves	LIF-1 to LIF-9
PEOPLE	Who acts, and what they may do	PPL-1 to PPL-9
DELIVERY	How content reaches humans	DEL-1 to DEL-9
PLATFORM	What it all runs on	PLT-1 to PLT-9

THE MAIN GAME

The Defense

Three to six players. Roughly 30 minutes. One player is the Arbiter each round; the role passes to the left when the round ends.

SETUP

- Shuffle the Term deck. Deal five Term cards to every player except the Arbiter.
- Shuffle the Doctrine deck and place it face-down within the Arbiter's reach.
- The Arbiter takes the Key. They do not play this round — they judge it.

THE ROUND

- **Flip.** The Arbiter turns over one Doctrine card and reads both the principle and its explanation aloud.
- **Choose.** Each player secretly picks ONE Term card from their hand that they believe best illustrates that doctrine. Play it face-down.
- **Reveal.** All at once. Nobody speaks yet.
- **Defend.** Going clockwise from the Arbiter's left, each player gets 30 seconds to argue why their term illustrates the doctrine. They must use the term correctly — not merely plausibly.
- **Challenge.** At any point during a defense, any other player may call CHALLENGE if they believe the term is being misused. See below.
- **Award.** The Arbiter gives the Doctrine card to the best argument. That is one point. Everyone draws back up to five.

> **The Challenge — this is where the learning happens**
>
> Call CHALLENGE and state precisely what is wrong with the usage. The Arbiter reads the relevant entry from the Key aloud — always aloud, always in full. If the challenge is correct, the challenged card is discarded with no chance at the doctrine, and the challenger scores one point immediately. If the challenge is wrong, the challenger loses one point and may not challenge again this round.

WINNING

First to five points. Doctrine cards won stay face-up in front of you; challenge points are tracked on paper. If two players reach five in the same round, the one holding more Doctrine cards wins — defending beats sniping.

TWO MORE WAYS TO PLAY

Variants

VARIANT ONE · NOMENCLATURE

A faster, meaner game about the fact that this vocabulary does not agree with itself. Deal the whole Term deck out evenly. On your turn, play one card face-up and name another term you claim it is confused with in practice — then say why in one sentence.

- If the Key's entry for either term names the other, or lists it as an alias, or flags the collision: score 2 and keep the card.
- If it does not, discard the card and score nothing.
- Any player may pre-empt you by calling the pair first if they think you are about to score — they take the 2 instead, but lose 3 if wrong.
- Highest score when the last card is played.

There are sixteen documented pairs in the Key. Students who have played this twice stop saying 'versioning' when they mean 'version control.'

VARIANT TWO · BUILD THE MODEL

The cooperative-feeling one, and the best for a classroom. The Arbiter reads a Scenario strip aloud. Every player races to lay down the SMALLEST valid set of Term cards that would model or solve it, then defends the set.

- Deal seven Term cards each. Arbiter reads a scenario.
- Players lay sets face-up as they are ready. Order matters — first down with a valid set wins ties.
- Smallest valid set scores 3. Any other valid set scores 1. An invalid set scores nothing and costs you the cards.
- The Arbiter has final say but must justify a rejection from the Key.
- Play all twelve scenarios, or to 15 points.

Suggested answers are printed beside each scenario, but treat them as a floor rather than a ceiling. A student who lays down a different set and defends it well has done something better than matching the answer key.

A note for whoever is teaching

The Arbiter role is the most valuable seat at the table, not the least. Rotate it deliberately and make sure your weakest students sit in it — reading definitions aloud while adjudicating other people's arguments is the single highest-leverage thing in this box.

CUT INTO TWELVE STRIPS

Scenarios

For Variant Two. Suggested sets are printed in grey — fold or trim them off if you would rather the Arbiter not see them.

01 A movie database must reject a release year of 3,000 — but accept 1994.
Attribute + Datatype + Validation Rule

02 The '2016 Annual Report' needs to be created. 'Managed Document' already exists.
Content Type + Content Object

03 Bob may edit the Privacy Policy, but the change must not go live until Alice signs off.
Permission + Approval + Role

04 One uploaded photograph must appear at five different sizes across the site.
Rendition + Delivery

05 Editors are told to stop typing. The final import has run; the new site has not launched.
Content Freeze + Content Migration

06 A checkbox reading 'Show Right Sidebar' needs a home. It affects one layout only.
Metadata + Attribute

07 Editors need a 'To Do' note on each article that readers must never see.
Metadata (editorial) + Editor

08 Press releases must appear newest-first, forever, without anyone maintaining a list.
Aggregation + Content Type

09 Editors work inside the same running system where visitors read the content.
Coupling Model (coupled) + Environment

10 A mobile app must pull content on demand. There is no web front end at all.
Coupling Model (headless) + API + Channel

11 A campaign page must vanish at midnight on the 31st without anyone being awake.
Scheduling (expiration) + Publication

12 Four unrelated companies will share one installation and its computing resources.
Tenancy (multi-tenant) + Environment

STRUCTURE STR-1

Content Type

ALSO CALLED
content template; content class

DON'T CONFUSE WITH
Template — and Content Object

TERMS OF ART · CM101

STRUCTURE STR-2

Content Object

ALSO CALLED
content; item

DON'T CONFUSE WITH
Content Type — the object is the instance, not the specification

TERMS OF ART · CM101

STRUCTURE STR-3

Content Element

ALSO CALLED
widget; block

DON'T CONFUSE WITH
Page — the difference is URL addressability

TERMS OF ART · CM101

STRUCTURE STR-4

Attribute

ALSO CALLED
property; field; parameter

DON'T CONFUSE WITH
Datatype and Validation Rule — an attribute merely holds the value

TERMS OF ART · CM101

STRUCTURE STR-5

Datatype

DON'T CONFUSE WITH
Validation Rule — the datatype is the coarse constraint

TERMS OF ART · CM101

STRUCTURE STR-6

Content Model

ALSO CALLED
structured content

DON'T CONFUSE WITH
Unstructured Content

TERMS OF ART · CM101

STRUCTURE STR-7

Metadata

ALSO CALLED
editorial metadata

DON'T CONFUSE WITH
Attribute — metadata is a role the data plays, not a storage slot

TERMS OF ART · CM101

STRUCTURE STR-8

Relationship

ALSO CALLED
reference; dependency; referential integrity

DON'T CONFUSE WITH
Aggregation — which can create relationships by inference

TERMS OF ART · CM101

STRUCTURE STR-9

Validation Rule

DON'T CONFUSE WITH
Datatype — the rule is the fine-grained constraint

TERMS OF ART · CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

LIFECYCLE LIF-1

Draft

ALSO CALLED
in draft

DON'T CONFUSE WITH
Versioning — 'draft' is often just a version label

TERMS OF ART · CM101

LIFECYCLE LIF-2

Approval

ALSO CALLED
approval chain; serial approval; parallel approval

DON'T CONFUSE WITH
Workflow — and Permission, which can achieve the same effect

TERMS OF ART · CM101

LIFECYCLE LIF-3

Workflow

ALSO CALLED
steps; activities; tasks

DON'T CONFUSE WITH
Approval — approval is one use of workflow, not its definition

TERMS OF ART · CM101

LIFECYCLE LIF-4

Versioning

ALSO CALLED
version history; version stack; version label; rollback; diff

DON'T CONFUSE WITH
Version Control — the two are often used interchangeably, wrongly

TERMS OF ART · CM101

LIFECYCLE LIF-5

Changeset

ALSO CALLED
project; edition

DON'T CONFUSE WITH
Scheduling — changesets are often scheduled, but are not scheduling

TERMS OF ART · CM101

LIFECYCLE LIF-6

Scheduling

ALSO CALLED
expiration

DON'T CONFUSE WITH
Archival — both remove content, by different means

TERMS OF ART · CM101

LIFECYCLE LIF-7

Content Freeze

DON'T CONFUSE WITH
Content Migration — the freeze is a phase of the migration

TERMS OF ART · CM101

LIFECYCLE LIF-8

Archival

DON'T CONFUSE WITH
soft delete — conceptually different; and Scheduling

TERMS OF ART · CM101

LIFECYCLE LIF-9

Publication

ALSO CALLED
unpublishing

DON'T CONFUSE WITH
Delivery — publication is the act, delivery the machinery

TERMS OF ART · CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

PEOPLE PPL-1

Editor

ALSO CALLED

power editor; approver; UGC editor; community manager; marketer

DON'T CONFUSE WITH

Administrator — though one person often holds both

TERMS OF ART · CM101

PEOPLE PPL-2

Administrator

DON'T CONFUSE WITH

Editor and Developer — roles overlap in practice

TERMS OF ART · CM101

PEOPLE PPL-3

Developer

ALSO CALLED

front-end developer; back-end developer; server-side developer

DON'T CONFUSE WITH

Administrator — configuration versus code

TERMS OF ART · CM101

PEOPLE PPL-4

Stakeholder

DON'T CONFUSE WITH

Editor — the stakeholder may never touch the system

TERMS OF ART · CM101

PEOPLE PPL-5

Role

DON'T CONFUSE WITH

User Group — sometimes functionally identical, sometimes not

TERMS OF ART · CM101

PEOPLE PPL-6

User Group

DON'T CONFUSE WITH

Role — the distinction varies by system

TERMS OF ART · CM101

PEOPLE PPL-7

Permission

ALSO CALLED

access control entry; authorization

DON'T CONFUSE WITH

Access Control List — the entry versus the collection

TERMS OF ART · CM101

PEOPLE PPL-8

Access Control List

ALSO CALLED

ACL

DON'T CONFUSE WITH

Permission — an ACL is a collection of them

TERMS OF ART · CM101

PEOPLE PPL-9

Authentication

DON'T CONFUSE WITH

Authorization — the single most common confusion in this suit

TERMS OF ART · CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

DELIVERY DEL-1

Delivery

DON'T CONFUSE WITH

Publication — and Coupling Model, which determines who does it

TERMS OF ART · CM101

DELIVERY DEL-2

Channel

ALSO CALLED

multi-channel; omni-channel

DON'T CONFUSE WITH

Rendition — both produce variants, for different reasons

TERMS OF ART · CM101

DELIVERY DEL-3

Template

ALSO CALLED

rendering; rendering template; output template

DON'T CONFUSE WITH

Content Type — in some systems 'template' means the type itself

TERMS OF ART · CM101

DELIVERY DEL-4

Templating Language

ALSO CALLED

token; control structure; flow control

DON'T CONFUSE WITH

Template — the language versus the artifact written in it

TERMS OF ART · CM101

DELIVERY DEL-5

Rendition

DON'T CONFUSE WITH

Versioning — renditions are NOT versions

TERMS OF ART · CM101

DELIVERY DEL-6

Page

ALSO CALLED

page-based CMS

DON'T CONFUSE WITH

Content Element — the page has a URL; the element does not

TERMS OF ART · CM101

DELIVERY DEL-7

URL Addressability

DON'T CONFUSE WITH

Page and Content Element — this is the property that separates them

TERMS OF ART · CM101

DELIVERY DEL-8

Personalization

ALSO CALLED

known personalization; anonymous personalization

DON'T CONFUSE WITH

AB Testing — both vary what a visitor sees

TERMS OF ART · CM101

DELIVERY DEL-9

Aggregation

ALSO CALLED

list; collection; menu; taxonomy

DON'T CONFUSE WITH

Relationship — aggregation can create relationships by inference

TERMS OF ART · CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

PLATFORM PLT-1

Repository

ALSO CALLED
repo

DON'T CONFUSE WITH
itself — it means both a content store and a code store

TERMS OF ART · CM101

PLATFORM PLT-2

Environment

ALSO CALLED
hosting; development; testing; staging; production

DON'T CONFUSE WITH
Staging — which is one kind of environment; and Tenancy

TERMS OF ART · CM101

PLATFORM PLT-3

Staging

ALSO CALLED
virtual staging

DON'T CONFUSE WITH
Environment — staging is a purpose, not a separate concept

TERMS OF ART · CM101

PLATFORM PLT-4

Coupling Model

ALSO CALLED
coupled; decoupled; headless

DON'T CONFUSE WITH
Delivery — the coupling model determines who performs delivery

TERMS OF ART · CM101

PLATFORM PLT-5

Application Programming Interface

ALSO CALLED
API; event model; events; triggers; event handlers

DON'T CONFUSE WITH
Connector and Plug-In — the API is what they are built with

TERMS OF ART · CM101

PLATFORM PLT-6

Connector

DON'T CONFUSE WITH
Plug-In — a connector is a class of plug-in

TERMS OF ART · CM101

PLATFORM PLT-7

Plug-In

ALSO CALLED
extension; module; component; add-on

DON'T CONFUSE WITH
Connector — which is a specific kind of plug-in

TERMS OF ART · CM101

PLATFORM PLT-8

Tenancy

ALSO CALLED
single-tenant; multi-tenant

DON'T CONFUSE WITH
Environment — tenancy describes who shares one

TERMS OF ART · CM101

PLATFORM PLT-9

Content Migration

ALSO CALLED
migration

DON'T CONFUSE WITH
Content Freeze — and Import, the mechanism it uses

TERMS OF ART · CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

TERMS OF ART

TERM

CM101

DOCTRINE D-01

Content and data are not the same thing

Content demands specific functionality, such as versioning, permissions, editorial tooling. Data does not require this. The difference between what a system calls content and data is often defined by the services applied to it.

DOCTRINE D-02

Code and content are not the same

Code operates on content, for logical processing and presentation. The two are often released in parallel to achieve objectives. However, in most cases, they are developed and managed separately.

DOCTRINE D-03

Content management is about managing change

We need content management because things change. If our content was static and never changed, would we need to manage it? Our needs for management are proportional to the amount of change present in the editing and presentation of the content.

DOCTRINE D-04

Content structure is a balance between granularity and usability

There is a temptation to structure content as tightly as possible, but in doing so we can inadvertently increase the work required to create, manage, and deliver it. Our goal should be to find the right balance between exactness and usability.

DOCTRINE D-05

Sometimes value comes from NOT managing something

Managing a specific unit of content always imposes overhead and can increase risk. In many situations it is to your benefit to exclude some content from exposure to humans by preventing its management and increasing the friction required to change it.

DOCTRINE D-06

Content almost always exists in relation to other content

Very rarely does a content object exist in isolation. It is almost always part of an aggregation of other content, and its relationship to that other content — its context — often defines behavior, appearance, and functionality.

DOCTRINE D-07

Content is delivered in artifacts

In some senses content is an abstract ideal. It exists as an idea, a message, or some unit of knowledge. It is used to create artifacts, which are the actual, concrete media that humans consume. One unit of content can generate multiple artifacts.

DOCTRINE D-08

Organizing content often also creates content

Aggregations of content can become content in and of themselves. Organizing content creates value, and that organization often needs the same services that the constituent content needs.

DOCTRINE D-09

There are different levels of content structure

Content can be structured inside a single property of an object, among multiple properties of the same object, an object in relation to other objects in the same repository, or even in relation to objects in other repositories.

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

TERMS OF ART

DOCTRINE

CM101

DOCTRINE D-10

Content management is about imposing boundaries and limits

Managing content means imposing rules. These rules restrict flexibility in service of greater functionality and value. A system with no limits and total flexibility is not managed in any sense.

DOCTRINE D-11

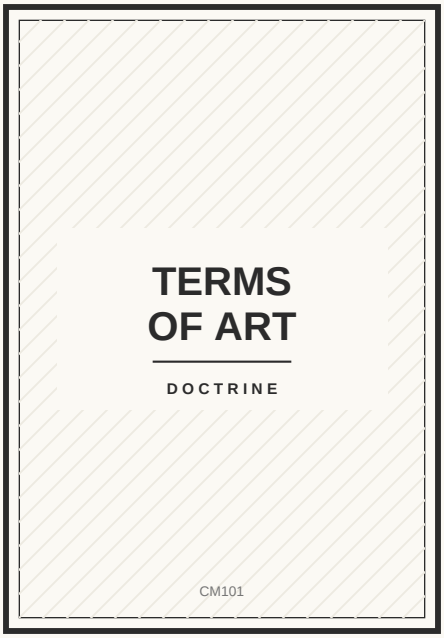
A CMS is a packaging of patterns and opinions

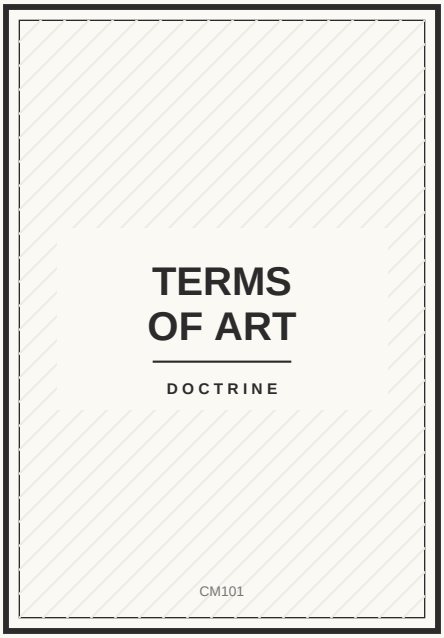
There is no grand unified theory of content management. A content management system is simply a packaging of the collected experiences and opinions of its designers. Consequently, they can differ markedly from one another.

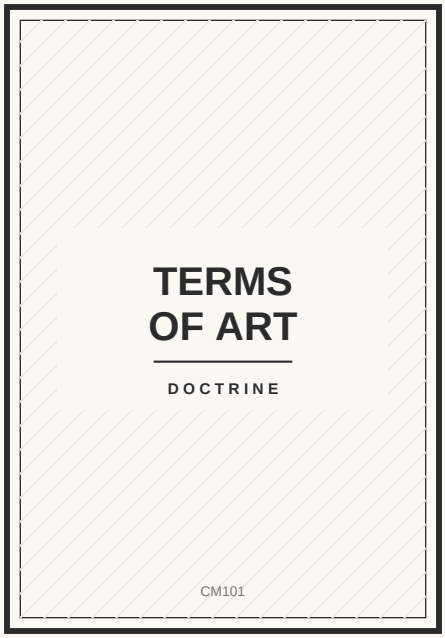
DOCTRINE D-12

Content is consumed much more often than it is created

Content is WORM-ish: write once, read many. This means it is efficient to do logical work during artifact creation that is not represented in the underlying content. This requires that the concept of an artifact be something repeatable and ultimately disposable.


TERMS
OF ART
DOCTRINE
CM101


TERMS
OF ART
DOCTRINE
CM101


TERMS
OF ART
DOCTRINE
CM101

THE ARBITER HOLDS THIS

The Key

Definitions for all 45 terms, grouped by suit. Read the relevant entry aloud when a challenge is called — the reading aloud is not ceremony, it is the mechanism.

STRUCTURE — How content is shaped

STR-1 Content Type

also called: content template; content class

The specification of a logical type of content, such as a News Article or a Blog Post. It defines the nature of a particular content object: the attributes it contains, how it is handled for approval and permissions. In modeling terms, a package of attributes; the values of those attributes constitute a content object. Nearly every content object must have a type — it has to know what it is.

STR-2 Content Object

also called: content; item

A single piece of managed content, often called simply 'content.' In simpler web-only systems it is concurrent with a page; in others it is generic — a collection of attribute values usable in any context. The suffix 'object' exists to distinguish the instance from its defining type: the '2016 Annual Report' object is based on the 'Managed Document' type.

STR-3 Content Element

also called: widget; block

A content object with no URL addressability, intended for embedding in other content objects or composing more complicated content. In many systems these are widgets or blocks; in others they are simply the norm, because there is no system-wide concept of a URL-addressable object. Distinguished from a page, which is meant to be accessed directly from a browser.

STR-4 Attribute

also called: property; field; parameter

The smallest unit of content and the basic building block of a content type. An attribute holds a value. At their simplest, content types are a collection of attributes: Title, Subtitle, Byline, Date Published, Body. Attributes usually have a datatype restricting what they can hold, and often validation rules limiting it further. Nomenclature varies widely and there is no standard.

STR-5 Datatype

A specification for the type of data which can be stored in an attribute. Common datatypes are Text, Number, Date, and Content Reference. A datatype naturally imposes one or more inherent validation rules to ensure entered data conforms to it. 'Number' would permit 3,000; only a validation rule would reject it.

STR-6 Content Model

also called: structured content

The collection of content types, attributes, relationships and datatypes in place to accurately describe a logical domain of content. Types are modeled two ways: discrete (the self-contained internal structure of the type) and relational (how an object relates to other objects and types). Usually not a formal document but the known collection of configuration or code establishing what is available.

STR-7 Metadata

also called: editorial metadata

Literally data about data; generally data which supports or describes other data. A vague term that may have lost some meaning — note there is no opposite term other than just 'data.' A 'Show Right Sidebar' checkbox is arguably metadata, being germane to one presentation rather than the content itself. Editorial metadata is never published but supports the editorial process, such as a 'To Do' field.

THE ARBITER HOLDS THIS

The Key, continued

STRUCTURE — How content is shaped

STR-8 Relationship

also called: reference; dependency; referential integrity

An explicit or implicit link between two content objects. It may be a reference from an attribute, an HTML link in rich text, a geographic relationship in a content tree, or an inferred relationship by co-existence in an aggregation. Explicit relationships live in an external structure such as a menu; implicit relationships are known internally to at least one object, as a child knows its parent.

STR-9 Validation Rule

A unit of logical processing ensuring entered data conforms to a type and format correct for its attribute. Implicit rules are imposed by the datatype. Explicit rules further restrict data based on the content domain: a movie database's Year attribute is a number, but a validation rule requires it fall between 1890 and roughly the current year.

LIFECYCLE — How content moves

LIF-1 Draft

also called: in draft

A content version not yet published but containing changes from the published version, if one exists. It usually does not refer to content that was published then expired or unpublished, only to content an editor expects to publish in future. Discarding a draft usually means the version is deleted without being published.

LIF-2 Approval

also called: approval chain; serial approval; parallel approval

A workflow process by which content submitted for publication is reviewed and approved by one or more people. Usually serial: A approves, which queues it for B. Less common is parallel, where a group receives it and some threshold approves. Often achieved with a workflow engine, but many systems have simpler dedicated approval systems, and approval can be effected purely through permissions.

LIF-3 Workflow

also called: steps; activities; tasks

A framework to move content through a defined set or map of specific steps, activities, or tasks. Content can only be in one such state per workflow at a time; when that state resolves, content moves to another and continues until the workflow ends. What constitutes a step is highly configurable — steps may change content or execute arbitrary code.

LIF-4 Versioning

also called: version history; version stack; version label; rollback; diff

The process by which changes to content are saved as a new copy instead of overwriting the existing state. A single instance at a moment in time is a version; the collection is the version history or stack. A version label designates a particular version as 'Published' or 'Draft.' Versioning lets editors roll back and audit changes for security or regulatory purposes.

LIF-5 Changeset

also called: project; edition

A grouping of pending content changes which can be reviewed and approved as a single unit, publishing at the same time, either on demand or scheduled.

LIF-6 Scheduling

also called: expiration

The unattended publishing or removal of a content version at a scheduled future point. This might be a change to an existing object (a new version) or the publishing of an entirely new object. Automatic removal is normally called expiration. Changes may be grouped into changesets and scheduled to publish as a unit.

The Key, continued

LIFECYCLE — How content moves

LIF-7 Content Freeze

A period during the late stages of a content migration when editors are asked not to make content changes. During this time a final content import has occurred, but the new implementation has not launched.

LIF-8 Archival

The act of un-publishing content by removing it from publication without irrevocably deleting it. Conceptually different from a soft delete, which moves content to trash or a recycle bin. Archiving means different things depending on the system and context.

LIF-9 Publication

also called: unpublishing

The act of making content available to the intended audience as part of the editorial process. Content is created, edited, collaborated on, reviewed, approved, and finally published. Content can also be unpublished — a vague term meaning removed from public view, through expiration, archival, or another mechanism.

PEOPLE — Who acts, and what they may do

PPL-1 Editor

also called: power editor; approver; UGC editor; community manager; marketer

A user responsible for performing editorial process to create, edit, or otherwise manage content. The core user of a CMS; most functionality is built around them. Subtypes include the power editor with some administrative duties, the approver who only moves content through approval, the community manager handling user-generated content, and the marketer concerned with delivery and performance rather than authoring.

PPL-2 Administrator

A role responsible for managing the settings, users, permissions, and other configurable functionality of a CMS. This role manages the system itself, not the content within it. The person holding it is often also a power editor and occasionally one of the implementing developers.

PPL-3 Developer

also called: front-end developer; back-end developer; server-side developer

The role normally responsible for installing, configuring, extending, and templating a CMS implementation. Roughly separated into front-end developers, who work with HTML/CSS and templates, and back-end or server-side developers, who work with the CMS itself and the extensions required to fulfil project requirements.

PPL-4 Stakeholder

A role which does not interact directly with the CMS but is primarily concerned with the results it is intended to bring about. A CEO may never use the CMS but expects it to increase revenue, reduce costs, or both. Stakeholders are often involved in purchasing decisions.

PPL-5 Role

Either a designation on which security information can be applied, or an organizational term meaning a collection of responsibilities. In the security sense, roles interact with user groups to allow assignment of permissions. In the organizational sense, the role abstracts responsibilities away from a specific person: responsibilities belong to the role, though a particular person holds it at any given time.

PPL-6 User Group

A group of user accounts organized to allow aggregate assignment of permissions. Permissions are often assigned to groups rather than individuals to ease management. Sometimes permissions can be assigned to groups directly, making them functionally identical to roles; in other cases permissions attach to roles, which are then assigned to groups or individuals.

THE ARBITER HOLDS THIS

The Key, continued

PEOPLE — Who acts, and what they may do

PPL-7 Permission

also called: access control entry; authorization

The granting of the ability, to a specific user, to perform a specific action, on a specific content object: Bob may edit the Privacy Policy. Usually granted in aggregate — the Editors group has all permissions over all content. Permissions are often inherited from a parent object or folder. The technical name for a single permission is an access control entry; their aggregate effect is the discipline of authorization.

PPL-8 Access Control List

also called: ACL

A collection of access control entries applied to a content object. The ACL determines who can read, edit, and delete it. ACLs are often inherited or applied in aggregate: content in a specific repository location might use an ACL provided by that location. Moved elsewhere, it abandons the prior ACL and takes one from the new location.

PPL-9 Authentication

The security controls in place to verify human ownership of a specific user account, usually by username and password. Different from authorization, which concerns what an account may do assuming successful authentication. Authentication is solely concerned with correctly matching a user account with a human. The authentication system is often pluggable, allowing an external user database.

DELIVERY — How content reaches humans

DEL-1 Delivery

The processes and systems employed to transmit content to requesting users. This ranges from the web server itself to wider concepts such as content optimization and the method of publishing content into a deliverable state. A decoupled CMS delegates delivery to other systems; a coupled CMS performs management and delivery in the same system.

DEL-2 Channel

also called: multi-channel; omni-channel

A distribution outlet for content; a method in which content is consumed. The web is a channel; Facebook is another. If creation and management is the supply side of content, the channel is the demand side — where content is actually received and communicated to humans. Different templates may be applied to prepare content for output into different channels.

DEL-3 Template

also called: rendering; rendering template; output template

Very generally, a process that transforms content into another format — most often a mixture of presentational markup and templating code preparing content objects for delivery. Multiple templates can produce different presentations of the same object. Normally stored as a file in source code management and managed as code, with at least one template per content type. A genuine naming collision: in some systems 'template' means the content type.

DEL-4 Templating Language

also called: token; control structure; flow control

The functionality and logic constructs available to templates to form a presentation from a content object. Generally designed as a less capable version of the CMS's native language for security, simplicity, and stability. Examples include Razor, Twig, Smarty, Freemarker, and Velocity. At minimum a language allows token replacement; more advanced ones allow control structures for looping and branching.

THE ARBITER HOLDS THIS

The Key, continued

DELIVERY — How content reaches humans

DEL-5 Rendition

Often used in digital asset management to describe different media formats automatically derived from a source media item. An image file might be processed into five renditions at different sizes. Critically, these are not versions — they are all renditions of the same version, made available to editors.

DEL-6 Page

also called: page-based CMS

A content object in a web content management system intended to be accessed directly from a browser via a URL. In its purest form a page is distinguished from a more abstract content element by the assignment of a URL. In some systems a page is simply a container organizing content elements; in others it is a specific type with page-like attributes such as title and meta tags.

DEL-7 URL Addressability

The ability of a content object to be the direct target of an inbound URL request. Content objects representing pages are necessarily URL addressable and usually form the operative content object during templating. Content elements intended for embedding are usually not addressable.

DEL-8 Personalization

also called: known personalization; anonymous personalization

The practice of modifying website content specifically for the current visitor. Known personalization applies when the visitor's identity is definitively known, usually via authentication. Anonymous personalization detects demographic, environmental, and technical information and adapts content in real time; the actual identity may be unknown, with targeting to a demographic category such as 'value buyer' or 'job seeker.'

DEL-9 Aggregation

also called: list; collection; menu; taxonomy

An intentional grouping of content for organization or presentation. Types include a hierarchical content tree, lists or collections as flat ordered sequences, a category or tagging system, and menus which tend to be hierarchical and often include non-content items. Aggregations can be manual and curated, or dynamic — a list that searches and displays all press releases in reverse chronological order.

PLATFORM — What it all runs on

PLT-1 Repository

also called: repo

Generically, a storage location. In content management it means one of two things: the functional unit of a CMS that stores content objects, providing library services such as modeling, versioning, permissions, and audit trails; or a source code management system storing code. The raw term can mean either, depending entirely on context.

PLT-2 Environment

also called: hosting; development; testing; staging; production

The computing infrastructure on which a CMS is installed and accessed. The term is often prefaced by a qualifier explaining its purpose: development, testing, staging, production, disaster recovery. The process around installing, running, and making server software available is known as hosting. Where one environment is provided by a vendor and accessed by multiple independent customers, this is multi-tenancy.

The Key, continued

PLATFORM — What it all runs on

PLT-3 Staging

also called: virtual staging

A pre-production environment for either code or content to be reviewed prior to publishing to production. Decoupled systems often have a separate staging environment to which content is deployed and previewed. Coupled systems generally use a virtual staging model, previewing content in the production environment by using version labels. Code is often deployed to staging for QA before production.

PLT-4 Coupling Model

also called: coupled; decoupled; headless

The degree to which management functionality is bound to delivery functionality. A coupled CMS binds both into the same software — editors and visitors interact with different sides of one system. A decoupled CMS separates them into different environments; publishing transmits content to delivery. A headless CMS has editorial features but no delivery, exposing content via web service. Decoupled pushes; headless waits to be pulled.

PLT-5 Application Programming Interface

also called: API; event model; events; triggers; event handlers

The language tools available for developers to communicate with and control software from code. A CMS API usually lets a developer connect from their own code and manipulate content objects. A key part is the event model offered to hook into events in the CMS's inner workings: most systems raise events in response to internal activity, and developers configure their own event handlers to execute at those moments.

PLT-6 Connector

A generic term referring to a class of plug-in which connects a CMS to another system. Connectors to digital asset management systems are common, as are connectors to enterprise content management systems. These allow content to be shared to the CMS and, less commonly, the reverse. Connectors to marketing automation and CRM platforms are also common.

PLT-7 Plug-In

also called: extension; module; component; add-on

An extension to a CMS packaged for distribution and installation into other instances of that CMS. The ecosystem around a particular CMS often generates plug-ins and maintains a common repository of them.

PLT-8 Tenancy

also called: single-tenant; multi-tenant

The relationship of an organizational entity to a software installation, indicating whether more than one unrelated entity interacts with the same software. Most systems are single-tenant: one implementation designed for one organization, on-premise or via SaaS, owning its own computational resources. Multi-tenant systems have multiple organizations sharing one installation and its resources, necessarily vendor-hosted on a subscription basis.

PLT-9 Content Migration

also called: migration

The movement of content from an old CMS implementation to a new one. Content migrations occur after a new CMS is implemented and all content has to be moved across. Sometimes conflated with the more general 'migration,' which refers to the entire process of switching platforms including redesign and implementation. Specifically, a content migration is the movement of existing content objects once the new implementation is available.

FOR VARIANT ONE

The confusion table

Sixteen pairs that the source material itself flags as collisions, aliases, or routine mix-ups. In Nomenclature, naming any of these correctly scores.

THIS	IS CONFUSED WITH	WHY IT MATTERS
Versioning	**Version Control**	The glossary says outright these are often confused and used interchangeably.
Authentication	**Authorization**	Authentication matches a human to an account. Authorization is what that account may then do.
Datatype	**Validation Rule**	'Number' permits 3,000. Only the rule rejects it.
Rendition	**Version**	Five image sizes are five renditions of ONE version.
Archival	**Soft delete**	Archival un-publishes. Soft delete moves to the trash. Not the same.
Template	**Content Type**	In some systems 'template' means the type itself. A real collision.
Page	**Content Element**	The page has a URL. The element is embedded.
Attribute	**Property / Field**	Same thing. There is no standard, and the glossary says so.
Role	**User Group**	Sometimes functionally identical, sometimes strictly layered.
Permission	**Access Control List**	One entry versus the collection applied to an object.
Workflow	**Approval**	Approval is one use of workflow — often a separate, simpler system.
Content Type	**Content Object**	The specification versus the instance built from it.
Repository	**Repository**	It means both the content store and the source code store.
Publication	**Delivery**	Publication is the editorial act. Delivery is the machinery.
Coupled	**Decoupled / Headless**	Decoupled pushes content out. Headless waits to be pulled.
Connector	**Plug-In**	A connector is a class of plug-in, not a peer of it.