# LV 3.7: Sisällönhallinta- ja julkaisujärjestelmät

# Sisällönhallintajärjestelmien perusteet

# Mitä on sisältö? (Boiko, 2002)

- **Määritelmä (Boiko, 2002)**[1]

  **”Like data, *content* is also information but it retains its human meaning and context.”**

  - Content Is Not Data
  - Content Is Information Put to Use
  - Content Is Information Plus Data

- Kaikki *informaatio* ei ole *sisältöä*. Informaatio muuttuu *sisällöksi*, kun siihen liittyy mielekäs tulkinta (human meaning) ja asiayhteys (context)
- **Määritelmän tulkintaa:**
  - *Esimerkki: tieto jalkapallojoukkueen harjoitusajoista on **informaatiota**. Harjoitusajoista muodostettu PDF-dokumentti on **sisältöä.***
  - Esimerkki: yrityksen nykyinen visio ja missio ovat **informaatiota.** Yrityksen HTML-muotoinen esittelysivu, jossa visio ja missio esitellään, on **sisältöä**.

1 Boiko, B. (2002). Content Management Bible.

# Mitä on sisältö? (jatkuu)

- **Määritelmä: ContentWatch[2]**
  - *Informaatio* muuttuu *sisällöksi*, kun se muotoillaan siten, että se on yhdellä tai useammalla tavalla *käyttökelpoista*
    - *Sisällön arvo* perustuu yhdistelmään sen 1) soveltuvuudesta eri käyttöihin, 2) saavutettavuudesta/saatavuudesta, 3) käytöstä, 4) käyttökelpoisuudesta, 5) brändi-tunnistettavuudesta sekä 6) ainutlaatuisuudesta

2 http://www.contentwatch.com/what.html

# CMS - määrittelyä

- **Sisällönhallintajärjestelmä (CMS – Content Management System)** on sisällön muokkaamiseen, hallintaan ja julkaisuun erikoistunut järjestelmä.
- Määritelmiä:
  - Boiko (2002):

    ”Content, therefore is information that you tag with data so that a computer can organize and systematize its collection, management, and publishing. Such a system, a *content management system*, is successful if it can apply the data methodologies without squashing the interest and meaning of the information along the way.”

  - http://www.contentmanager.eu.com/history.htm

    **"A CMS is a tool that enables a variety of (centralised) technical and (de-centralised) non technical staff to create, edit, manage and finally publish (in a number of formats) a variety of content (such as text, graphics, video, documents etc), whilst being constrained by a centralised set of rules, process and workflows that ensure coherent, validated electronic content."**

# CMS:n keskeinen idea

- Keskeinen ajatus CMS:issä on, että *sisältö* voidaan ”kapseloida” yksiköihin
- Sisällönhallinnan perusyksikköä kutsutaan ***sisältöyksiköksi*** (*content unit*)
- Sisältöyksikkö koostuu varsinaisesta *sisällöstä* (data) sekä siihen liitetystä *metatiedosta* (metadata)
  - Sisältöyksiköihin jäsennettyä tietoa voidaan hallita; muokata, luokitella, kategorisoida
  - Metatieto mahdollistaa *automatisoidun* tietojenkäsittelyn. Metatieto voi olla joko järjestelmäkohtaista, tai yleisempää
- Se miten ja minkä kokoisiksi sisältöyksiköiksi tieto jäsennetään riippuu siitä, mitä sisällönhallintajärjestelmällä halutaan hallita

# Sisällönhallintajärjestelmän lisäarvot

Odotettavissa olevia sisällönhallintajärjestelmästä saatavia lisäarvoja:

- *Tuottavuuden parantuminen*: tekijät (sisällöntuottaja, graafinen suunnittelija, ...) voivat keskittyä omaan osaamisalueeseensa
- *Oikeuksien hallinta*: dokumentin elinkaaren vaiheisiin vaikuttavien henkilöiden täsmällinen määritteleminen ja rajaaminen
- *Työnkulun (workflow) ohjaus*: Dokumentin elinkaaren vaiheistaminen ja tekijöiden vastuualueiden selkeyttäminen. Esimerkkejä: luominen, muokkaaminen, tarkastaminen, hyväksyminen, julkaiseminen, ...
- *Yhtenäinen ulkoasu*: Tieto syötetään sisällönhallintajärjestelmään yhtenäisessä muodossa, joten myös sisällön julkaisumuodot yhtenäistyvät. Apuna yhtenäistämisessä ovat sivupohjat, tyylitiedostot & funktiokirjastot
- *Ulkoasun muokkaaminen:* ks. Yhtenäinen ulkoasu

*Huonon suunnittelun seurauksena tai cms:n valinnan epäonnistuessa tuloksena on jotain päinvastaista, esimerkiksi tuottavuuden heikentyminen, byrokratian lisääntyminen tai pakollisten esitietojen lisääntyminen. Esimerkkejä löytyy vaikkapa valtion sähköisistä tietojärjestelmistä*

# CMS-järjestelmien luokittelua

- Sisällönhallintajärjestelmiä voidaan luokitella erityisesti niiden sovellusalueen perusteella, esimerkiksi seuraavasti:
  - Mediasisällönhallintajärjestelmät
  - Enterprise-sisällönhallintajärjestelmät
  - Web-sisällönhallintajärjestelmät
  - Dokumenttienhallintajärjestelmät
  - Komponenttipohjaiset sisällönhallintajärjestelmät

# Esimerkki luokittelusta: web-sisällönhallintajärjestelmät

Web-sisällönhallintajärjestelmillä tarkoitetaan erityisesti HTML-muotoisen sisällön hallintaa erikoistuneita järjestelmiä (lyhenne usein Web CMS)

- Karkea jaottelu:
  - *Offline-käsittelyyn pohjautuvat*
    - Sisältöön nivotaan mallinteisiin (templates) ja lopputuloksena syntyy julkaisukelpoisia web-sivuja
    - Esimerkkejä: Adobe Contribute, Bricolage
  - *Online-käsittelyyn pohjautuvat*
    - HTML-sisältö generoidaan lennosta (tai haetaan välimuistista)
    - Esimerkkejä: WordPress, Drupal, Plone, Joomla!
  - *Online- ja offline-käsittelyn yhdistävät järjestelmät (Esim. Blosxom)*
- Kehityksen suuntana siirtyminen *online-pohjaisiin* järjestelmiin

## Toinen näkökulma luokitteluun: dataa vai dokumentteja?

Toinen näkökulma järjestelmien luokitteluun on luokittelu tiedon perusteella. Kaksi tiedon jäsennystapaa on tunnistettavissa:

- *Datakeskeinen tieto* on jäsennettävissä pieniin palasiin, jotka voidaan tallentaa esimerkiksi relaatiotietokantaa. Esimerkkejä: reseptit, levyn tiedot, henkilötiedot
- *Dokumenttikeskeinen tieto* muodostuu jotakin tiettyä asiaa käsittelevistä kokonaisuuksista. Tiedon tallentaminen onnistuu parhaiten dokumenttitietokantaan. Esimerkkejä: kirja, raportti, käyttöohje, oppimateriaali, …

- Sisällönhallintajärjestelmissä ensisijainen jäsennystapa on monesti dokumenttikeskeinen
  - Etu: sisältöyksiköt muodostavat suoraan loogisia kokonaisuuksia, mutta datakeskeinen tieto on usein *sirpaloitunut* sisältöyksiköiden välille tai *sekoittunut* muuhun tietoon
- Ensisijaisesti datakeskeisellä jäsennyksellä on omat moninaisuutensa:
  - Etu: datakeskeinen tieto on helposti hallittavissa, mutta dokumentit on usein *koostettava* joukosta sisältöyksiköitä

# Sisällönhallintajärjestelmien ominaisuudet

# CMS: Vastuut ja ominaisuudet

Vaikka CMS:ien ominaisuudet vaihtelevat toimialuettain, voidaan seuraavat yleiset ominaisuudet tunnistaa (Lowdnes & Browning, 2001[3]):

- ***Sisällön tuottaminen (authoring):*** sisällön tuottaminen ja syöttäminen
- ***Työnkulun hallinta (workflow):*** sisällön (hallinnolliset) käsittelyn vaiheet sisällöntuotannosta julkaisuun
- ***Sisällön varastointi (storage):*** tuotetun sisällön varastointi tietovarastoon (repository)
- ***Julkaiseminen (publishing):*** tiedon julkaiseminen tietovarastosta eri muotoihin

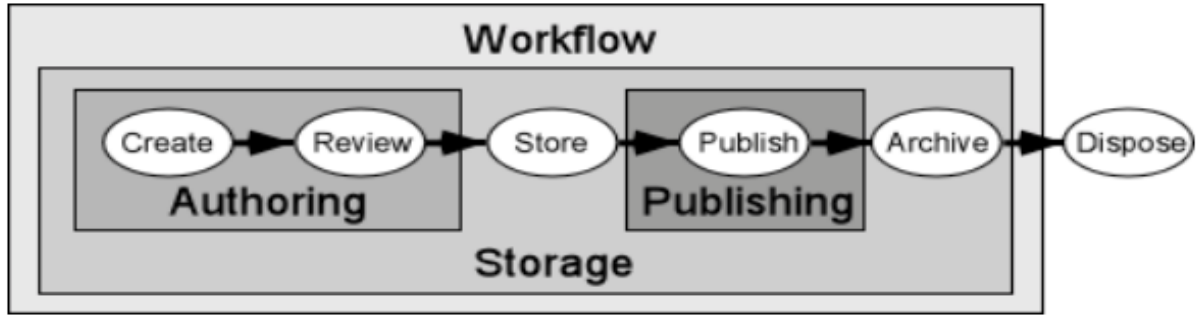



*Illustration 1: Toiminnallinen näkökulma CMS-järjestelmiin (Lowdnes & Browning, Ort & Vidgen et al. mukaan)*

3 Lowdnes, M. & Browning, P. JISC TechWatch Report: Content Management Systems

## Sisällön tuottaminen/tuominen

- Jotta tietoa voidaan hallita sisällönhallintajärjestelmillä, on se jollain tavalla kyettävä tuomaan järjestelmään tai vaihtoehtoisesti tuottamaan se sinne suoraan järjestelmän tarjoamilla välineillä

## Reunaehtoja sisällön luomiselle

- Sisällön tuomiseen/tuottamiseen liittyy usein seuraavia reunaehtoja:
  - *Sisällön ja ulkoasun erottaminen*: keskeinen idea tiedon *rakenteistamisessa*, joka taas on pakollinen reunaehto monikanavajulkaisemiselle
  - *Hajautettu muokkaaminen:* sisällöntuottajia voi olla useita, joten CMS:n on mahdollistettava esimerkiksi tiedon lukitseminen ja versiointi
  - *Uudelleenkäyttö:* sama tieto - esimerkki yksittäinen dokumentti tai tekstikappale - voidaan julkaista useassa eri jakelukontekstissa
  - *Metatiedon tuottaminen:* sisällön hallinta edellyttää metatiedon keräämistä, joka ei kuitenkaan saa tukehduttaa sisällöntuotantoprosessia
  - *Linkittäminen:* sisällöntuottajan on voitava määritellä sovelluksen sisäisiä linkkejä, muutokset rakenteessa ja monikanavaisuus on otettava huomioon
  - *Käytön helppous ja tehokkuus: välttämätöntä, mutta toteuttaminen todellinen haaste (vrt. Web-lomakkeet ja sisällöntuottajien WYSIWYG-tausta => Word)*

## Sisällön tuominen (import) (1/2)

- Sisällön tuomisella (import) tarkoitetaan erityisesti sellaisen sisällön tuomista sisällönhallintajärjestelmään, joka tallennusmuodoltaan (formaatiltaan) ei suoraan ole järjestelmän tuntemassa muodossa
- Mahdollisuus sisällön tuomiseen on usein oleellinen osa erityisesti siirryttäessä sisällönhallintajärjestelmästä toiseen
- Sisällön tuominen voidaan usein tehdä joko käsisyötteisesti tai ohjelmallisen automatisoinnin mahdollistavan rajapinnan (API) kautta
- Yhdessä ohjelmarajapinnan kanssa tuomisprosessia voidaan automatisoida myös mm. skriptejä käyttämällä. Osassa järjestelmissä käyttöönottoa tuetaan erityisillä käyttöönotto (migration) skripteillä

## Sisällön tuominen (import) (2/2)

- Tuontiprosessissa lähdemuotoinen tiedosto muunnetaan sisällönhallintajärjestelmän tukemaan muotoon, noudattaen käytettyä *tieto/metatieto-jakoa*
- On huomattava, että tuontiprosessi ei välttämättä ole häviötön: tietoa saatetaan muokata siten, että järjestelmään tuotu tieto ei ole palautettavissa enää alkuperäiseen muotoonsa
  - Tämä ongelma tulee vastaan erityisesti jos sisällönhallintajärjestelmä käyttää alkuperäisistä sisältöyksiköistä poikkeavaa tieto/metatieto-jaottelua
- *Dokumenttienhallintajärjestelmät* ovat erityisesti sellaisia sisällönhallintajärjestelmiä, jotka pyrkivät tukemaan sisällön (dokumenttien) häviötöntä, alkuperäisen muodon säilyttävää hallintaa

## Sisällön tuottaminen

- Järjestelmästä riippuen, myös sisällön tuottamista voidaan kokonaan tai osittain tehdä itse järjestelmässä.
- Erityisesti web-sisällönhallintajärjestelmille on erittäin tyypillistä, että sisältö voidaan myös tuottaa suoraan järjestelmässä
  - Esimerkiksi web-dokumenttien tuottamiseen tarjotaan WYSIWYG-editori
- Vastaavasti osassa järjestelmiä tuotanto on usein selvästi eriytetty ulkopuolelle:
  - Esimerkkinä dokumenttien-, ja medialementtienhallintajärjestelmät

## Työnkulun hallinta ja tiedon varastointi

- Järjestelmään tuotu tieto varastoidaan järjestelmän sisäisen tallennustavan mukaisesti. Varastoidulle tiedolle voidaan soveltaa työnkulun hallinnan keinoja.

# Tiedon varastoinnin toteutus

- **Tietovarasto** voidaan käytännössä toteuttaa monella eri tavalla:
  - ***Määrämuotoinen (teksti)tiedosto* (flat file)**: Tiedot tallennetaan tekstitiedostoon esimerkiksi Comma Separated Value (CSV) -muodossa. Osa ASCII-merkeistä on varattu tietojen jäsentämiseen (mutta loputkin voi saada käyttöön pakomerkeillä)
  - ***Rakenteinen dokumentti:*** Tallennusmuotona käytetään esimerkiksi jotakin XML-kieliopin mukaista merkkauskieltä. Mahdollistaa tietojen suoraviivaisen siirtämisen ja uudelleenkäytön
  - ***Relaatiotietokanta:*** Tiedot tallennetaan relaatiotietokannan tauluihin riveinä. Erilaisiin tekstitiedostoihin verrattuna merkittävästi tehokkaampi ja siten skaalautuvampi vaihtoehto
  - ***Dokumenttitietokanta:*** Järjestelmään tuodut dokumentit, esimerkiksi mediaelementit, voidaan tallentaa sellaisenaan dokumenttitietokantaan. Myös osa relaatiotietokannoista tukee vastaavaa toiminnallisuutta.
- Relaatiotietokanta on (usein) suositeltavin vaihtoehto, kun tarkoituksena on toteuttaa sisällönhallintajärjestelmä suurelle joukolle käyttäjiä. Tekstitiedostoihin perustuvien vaihtoehtojen etuna on ***parempi siirrettävyys***.

## Työnkulun hallinta

- Työnkulun hallinnalla (workflow) tarkoitetaan niitä keinoja, joilla sisältöyksiköitä voidaan hallita niiden elinkaaren aikana.
- Käsitteellisellä tasolla ainakin seuraavat vaiheet ovat usein tunnistettavissa:
  - **Luonti:** sisällön tuonti/tuottaminen
  - **Hyväksyntä:** sisällön hyväksyminen ennen pitkäaikaisempaa säilytystä / julkaisua
  - **Varastointi:** tiedon pysyväisluonteinen varastointi
  - **Julkaisu:** tiedon julkaisu ulos järjestelmästä
  - **Arkistointi:** julkaisusta poistetun, mutta säilytettävän tiedon käsittely
  - **Tuhoaminen:** tiedon lopullinen hävittäminen

## Työnkulun hallinta: läpileikkaavat ominaisuudet

- **Versiointi ja arkistointi:** Versiointi mahdollistaa muutosten hallinnan, virheistä toipumisen ja tiedon palauttamisen. Arkistointi on usein parempi vaihtoehto kuin tiedon tuhoaminen
- **Työnkulun ohjaus:** Työnkulun on oltava muokattava, joustava ja riippumaton organisaatiossa tapahtuvista muutoksista
- ***Turvallisuus:*** pääsynhallinta, käyttöoikeudet, virheistä toipuminen, tietoturva
- **Integrointi ulkoisiin järjestelmiin:** CMS on ainoastaan yksi osa yrityksen tietojärjestelmää. Tarve tiedon kopiointiin tai mekaaniseen siirtämiseen estää tehokkaan toiminnan
- ***Raportointi*****:** Raportointi (mahdollisesti proaktiinen) järjestelmässä tapahtuvista virheistä (ylläpitäjä) tai esimerkiksi muutoksista sovelluksen tilassa (ylläpitäjä ja käyttäjä). Proaktiivinen tarkoittaa tässä sitä, että järjestelmä välittää raportit ilman käyttäjän aktiivista toimintaa

## Yleisiä huomioita työnkulusta

- Monessa tapauksessa työnkulun hallinnan piirteet ovat keskeinen osa siitä, miten sisällönhallinnanjärjestelmä soveltuu eri käyttötarkoituksiin
- Järjestelmät tukevat enemmän tai vähemmän myös itse työnkulun muokkaamista ja hallintaa.
  - Nyrkkisääntönä voidaan pitää, että mitä enemmän muokattavuutta on, sitä järeämpi järjestelmä on kyseessä
- On huomattava, että sisällönhallintajärjestelmien osalta työnkulun hallinnalla tarkoitetaan usein *teknisesti toteutettua, automatisointia tukevaa työnkulun hallintaa*.
  - Aina työnkulun hallintaa ei kannata toteuttaa kokonaan automatisoitujen, teknisten välineiden avulla, vaan joskus voi olla järkevää valita kevyempi ratkaisu, ja sopia hallinnasta *käytännöillä*

## Julkaiseminen

- Käytännössä melkeinpä kaikki sisällönhallintajärjestelmät tukevat tiedon automatisoitua julkaisemista jollain tasolla
- Julkaisuprosessin automatisointi tapahtuu *julkaisujärjestelmällä.*
  - Kevyt sisällönhallintajärjestelmä voi koostua oikeastaan pelkästään julkaisujärjestelmästä. Tästä syystä termejä julkaisujärjestelmä ja sisällönhallintajärjestelmä toisinaan näkee käytettävän synonyymeinä
  - Raskaammassa sisällönhallintajärjestelmässä julkaisujärjestelmä on selvästi erillinen osa järjestelmää

# Julkaiseminen: reunaehtoja

- Julkaisemiseen voi usein liittyä erilaisia reunaehtoja, jotka pitää huomioida:
  - *Ulkoasu*: ulkoasun irrottaminen esimerkiksi tyylitiedostojen avulla
  - *Sivupohjat:* Asettelun määritteleminen sivupohjilla, joiden muokkaaminen onnistuu ilman teknistä osaamista. Jopa WYSIWYG-käyttöliittymä sivupohjien muokkaamiseen?
  - *Laajennettavuus:* Järjestelmän on mahdollistettava jatkuva kehitystyö. Uusien julkaisemiseen liittyvien toimintojen toteuttamisen on oltava helppoa
  - *Julkaisumuodot:* Tiedon julkaiseminen eri muotoihin on oltava mahdollista. Mahdollisuus uusien julkaisumuotojen lisäämiseen on tärkeä
  - *Personointi:* tiedon julkaiseminen käyttäjän käyttäjäryhmän perusteella mukautettuna
  - *Käytön tilastointi*: Käytön lokidatan tallentaminen tietokantaan ja erilaisten raporttien muodostaminen. Esimerkkejä: suosituimmat sivut, päivittäinen käyttö tai hakutoiminnon käyttö (virheelliset haut => käytettävyyden edistäminen)

# Julkaisujärjestelmien luokittelua

Julkaisujärjestelmät voidaan jakaa toimintaperiaatteen perusteella kolmeen luokkaan:[4]

- *Yksi versio kaikille*: Hypermediasovelluksesta julkaistaan yksi versio, jonka on sovelluttava kaikkiin käyttötarkoituksiin. Tavoitteena *One size fits all*, tuloksena usein *One size fits nobody*
- *Räätälöity julkaiseminen*: Tässä lähestymistavassa julkaistaan lueteltu joukko versioita, joista valitaan kuhunkin käyttötarkoitukseen parhaiten sopiva versio. Valinta suoritetaan yleensä käyttäjän toimesta, mutta versio voidaan valita esimerkiksi asiakasohjelman (selain) toimittamien tietojen perusteella automaattisesti
- *Joustava julkaiseminen:* Sovelluksesta asiakkaalle toimitettavat näkymät räätälöidään pyynnön yhteydessä välitettyjen tietojen perusteella tiettyyn julkaisukontekstiin (delivery context) sopivaksi.

4 http://www.w3.org/TR/di-atdi/

## Esimerkki räätälöidystä julkaisemisesta

- *Kuvitteellinen esimerkki*: uutissähkeiden julkaisu samasta järjestelmästä kahteen eri kontekstiin: painettuun lehteen ja verkkoon
- Pohjana molemmille julkaisuille sama *sisältöyksikkö.* Sisältöyksiköissä toistuu seuraava rakenne:
  - *Päivämäärä:* julkaisupäivämäärä
  - *Otsikko:* sähkeen otsikko
  - *Sisältö:* sähkeen sisältö
  - *Lähde:* sähkeen lähde
- Sisältöyksiköihin voi liittyä myös muuta tietoa/metatietoa, mutta julkaisun yhteydessä *ne karsitaan pois*

## Esimerkki joustavasta julkaisemisesta

- *Kuvitteellinen esimerkki:* dokumenttien julkaisu päätelaitteen ja käyttäjäkohtaisen personoinnin perusteella
- Pohjana kaikille julkaisuille on sama *sisältöyksikkö*.
- Valitaan päätelaitteista esimerkiksi 2-5 erilaista vaihtoehtoa
- Lisäksi *personoidaan* sisältöyksikköä käyttäjäkohtaisesti: esimerkiksi lisätään sinne käyttäjän omia muistiinpanoja tai mukautetaan rakennetta käyttäjän personointiasetusten perusteella
- Päätelaitteiden ja personointiprofiilien yhdistelmiä on niin monta, että käytännössä joustava julkaisu on ainoa mahdollisuus
- Haasteena joustavasa julkaisemisessa on, että julkaisujärjestelmää on kyettävä käyttämään suoraan pyynnöstä

## Julkaisemisen käynnistäminen: manuaalisesti vai pyynnöstä?

- Järjestelmästä riippuen, julkaisu voidaan toteuttaa joko manuaalisesti, ajoitettuna eräajona tai suorittaa vasta sisällön hakupyynnön yhteydessä
- Manuaalisella / eräajo -julkaisulla voidaan toteuttaa myös räätälöityä tai monikanavajulkaisua.
  - Haasteena on, että räätälöintimahdollisuuksien/kanavavaihtoehtojen kasvaessa julkaisu voi tuottaa hyvinkin paljon tietoa
- Pyynnön yhteydessä (on-demand) julkaisussa sisällönhallintajärjestelmä suorittaa julkaisuprosessin vasta tietoa pyytäessä
  - Käytännössä vaatimus joustavalle julkaisemiselle. Etuna lisäksi mm. julkaisun vaivattomuus (ei tarvitse erikseen käskeä julkaisua).
  - Haasteena usein hitaus: ratkaistavissa osin mm. välimuistilla (cache)

## Tiedon vienti (export)

- Myös tiedon vienti järjestelmästä voidaan hahmottaa yhdenlaiseksi julkaisuprosessiksi:
  - Tarkoitus on, että vientiprosessin avulla sisällönhallintajärjestelmään taltioidut tiedot voidaan siirtää mahdollisimman toisiin järjestelmiin

    → Pyrkimys kaiken sisällönhallintajärjestelmän keräämään tiedon ja metatiedon säilyttämiseen
  - Vientiä voidaan soveltaa joko yksittäisten sisältöyksiköiden tai koko järjestelmän tasolla
    - Hyvin toteutetun viennin avulla tietovarasto voidaan siirtää saman järjestelmän toiseen versioon tai kokonaan eri sisällönhallintajärjestelmään

Puutteelliset vienti/tuonti-ominaisuudet ovat valitettavan yleisiä. Yksi syy voi olla, että toteutus ei ole ollut kannattavaa. Toisaalta voi olla jopa järjestelmän toimittajan edun mukaista, että sen vaihtaminen on hankalaa. Seurauksena riippuvuus yksittäisestä järjestelmästä ja sen toimittajasta (*vendor lock-in)*

## Lopuksi

- Sisällönhallintajärjestelmien käsitteistö tarjoaa mielekkään kehyksen erilaisten järjestelmien ominaisuuksien ja piirteiden kartoittamiseen ja vertailuun
  - Vaikka mm. järjestelmien toteutusteknologiat ja tarkat toiminnallisuudet vaihtelevat, voidaan CMS-käsitteistöllä mm. ymmärtää niiden välisiä eroja ja yhtäläisyyksiä
- Toisaalta CMS-käsitteistön ja ajattelun haasteena on se, että niissä ei juuri oteta kantaa toteutusyksityiskohtiin tai käytettävään mediaan
  - Erityisesti webissä voi paikoin olla hyväkin kyseenalaistaa se paradigma, jonka varaan sisällönhallintajärjestelmät usein rakentuvat
    - Mm. miten sosiaalinen media, wikit, blogit, web-syötteet, ym. web-keskeisemmät toimintatavat vaikuttavat CMS paradigman soveltamiseen?

# Demo: Drupal-sisällönhallintajärjestelmä

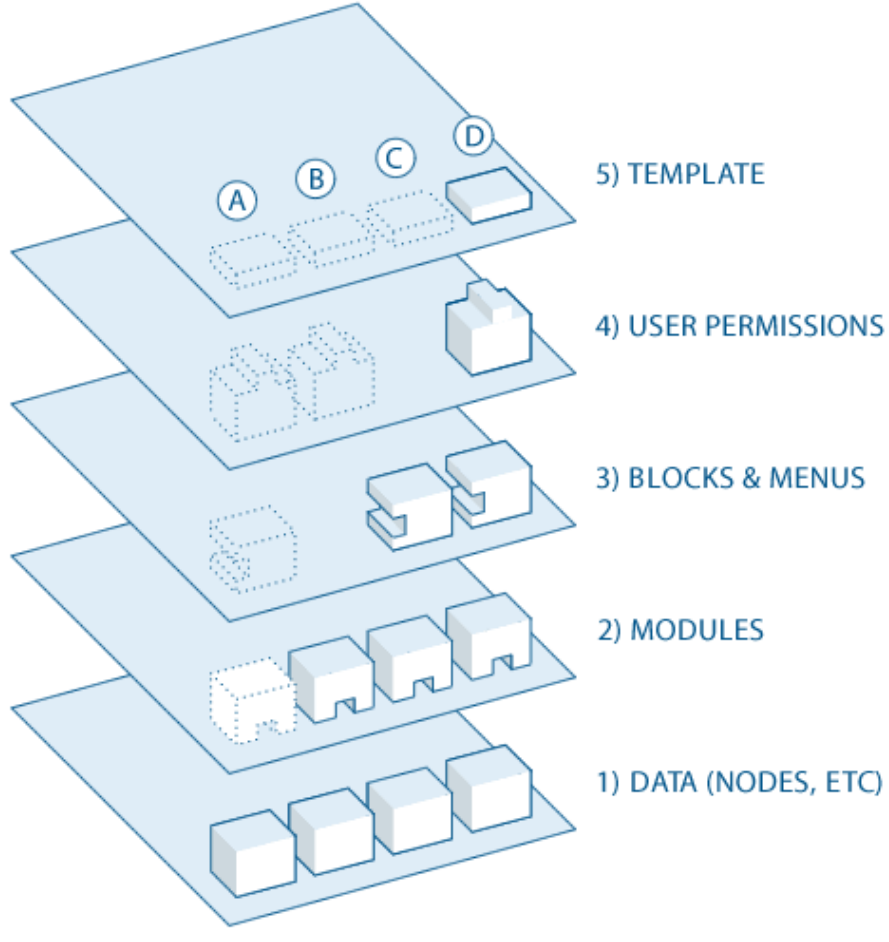
# Drupal

- Avoimen lähdekoodin sisällönhallintajärjestelmä/-alusta (https://drupal.org/)
  - Mukautuva alusta sisällönhallintajärjestelmien toteuttamiseen
  - Toimii jo sellaisenaan sisällönhallintajärjestelmänä
    - Saatavilla myös eri käyttötarkoituksiin räätälöityinä (distributions)
  - Käydään demossa läpi Drupal:n asentaminen ja käyttöönotto

# Asennus

- Asennuspaketin nouto (drupal.org)
- Sijoitus www-hakemistoon
- Uuden tietokannan + käyttäjän luonti:
  - phpMyAdmin: Users → Add user
- Asetusohjelman ajo (http://localhost/asennuskansio/)
  - Tietokanta
  - Sivuston asetukset (”site information”)
  - Done!

# Yleiskatsaus




http://drupal.org/getting-started/before/overview

# Toimintojen läpikäynti

- Admin-valikko
- Sisällön lisäys/muokkaaminen
- Rakenteen muokkaaminen
  - Lohkot (blocks)
  - Sisältötyypit (content types)
  - Valikot (menus)
  - Luokittelujärjestelmät (taxonomies)
- Ulkoasu (teeman vaihtaminen)
- Käyttäjät / käyttöoikeudet
- Moduulit