



Drupal Europe

Darmstadt, Germany

Sep 10 - 14, 2018

www.drupaleurope.org



Digital Transformation + Enterprise

TRACK SUPPORTED BY

AcquiaTM

Managing digital service data streams with **Migrate API**

13.09.2018 By Jari Nousiainen & Anu Määttä

00

Introduction

Introduction



Anu Määttä

Drupal Developer, Service Manager
@narnua on Twitter and drupal.org



Jari Nousiainen

Drupal Competence Lead
@jnous on Twitter
holist on drupal.org

Siili Solutions (@SiiliSolutions) is a digitalization consultancy based in Finland.

We have globally 660 employees and offices also in Poland, Germany and the United States.

Introduction

Building enterprise online services often requires some data stream management.

Drupal 8 provides tools for data stream management that suit different business needs.

We will present some use cases and outline how to design the solution for your requirements.

Contents

1. Data stream use cases
2. Migrate API overview
3. Case examples
4. Solution considerations

01

Data stream use cases

Data stream management

Enterprise websites need to incorporate data from other information systems.

Product information

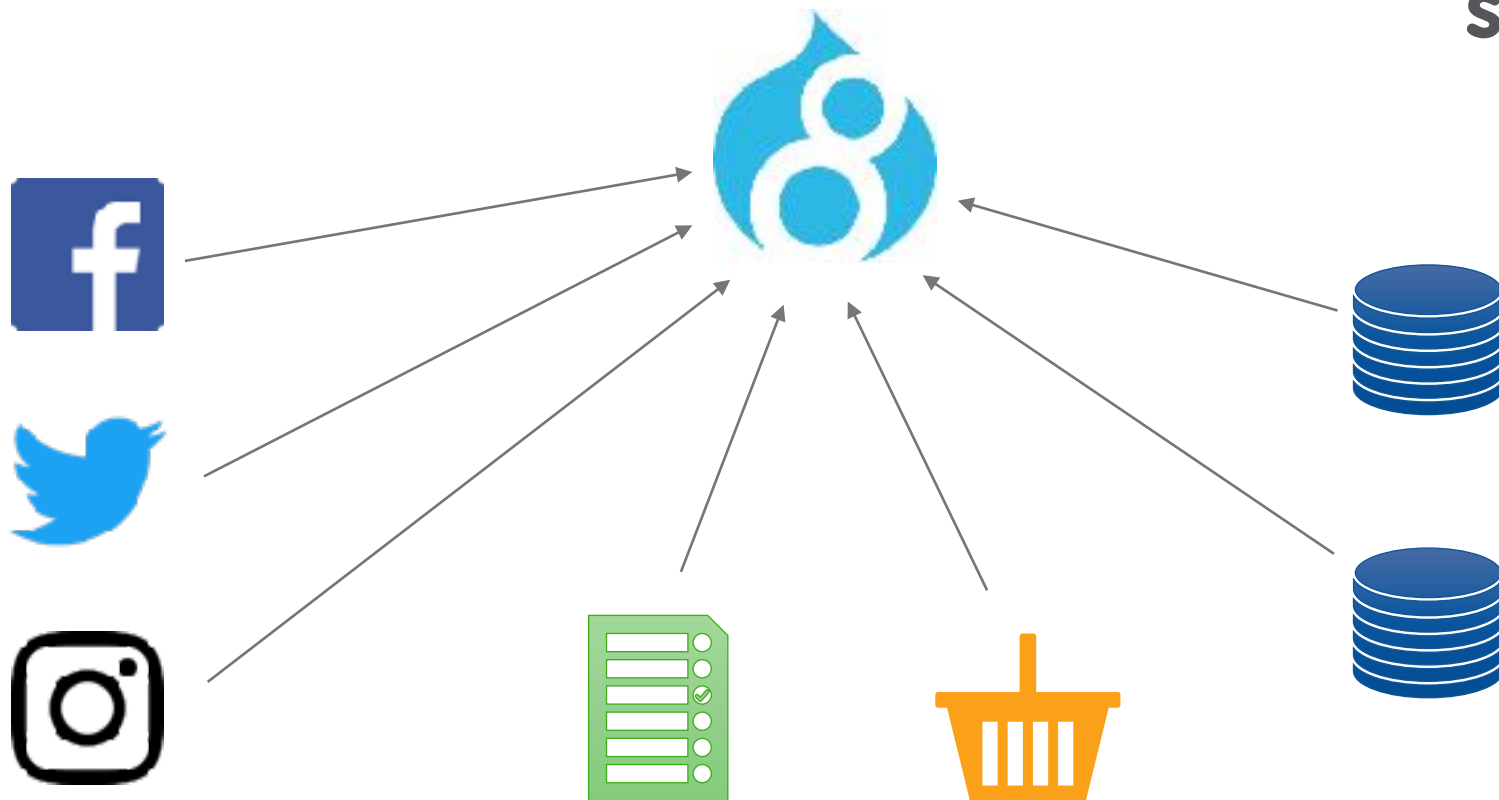
Service catalogue

E-commerce solutions

ERPs

Publishing platforms

Master data systems



Example cases

1. Exposing offering on public website
2. Showcasing social media presence, with styles
3. Managing data within enterprise architecture

Case 1: Publishing products or services

Case: A government agency provides services to the public and other officials

- Services need to be listed on the public site
- Prices for the services need to be listed
- Information is managed in a master data system
- JSON and XML interfaces are exposed
- Sales and billing goes through an another system

Case 2: Social media showcase

Case: Social media posts should be part of site newsfeed

- Display content posted on social media also on own site
- On company branded site, content should follow brand styles.
- Get social media content and site news content in one stream.
- Engagement needs to be direct.

Case 3: Really getting off the island

Case: Content needs to move outwards from Drupal

- Complex environments require more connectivity.
- Evolution of sites, sub-sites outgrowing their "mother-sites".
- Content sharing between different sites running different systems.
- REST API on the receiving end.
- Requires expertise on both ends.
- Modular approach to support different destinations.
- Serves as a "CMS exit strategy".

A built-in solution for handling data streams

Drupal 8 Migrate API can be used to solve all this.

A framework to fetch, process, save and update data

A system of reusable components

A versatile consumer of databases, export files and APIs.

02

Migrate API overview

What is Migrate API

The **Migrate API** provides services for migrating data from a source system to **Drupal 8**.

But it also works the other way around.

What does Migrate API do?

Migration is an ETL process

Extract, Transform, Load

Source, Process, Destination

Source

- Authentication
- ERP
- Legacy system
- Master data storage
- Database
- REST API
- XML, CSV, JSON exports

Process

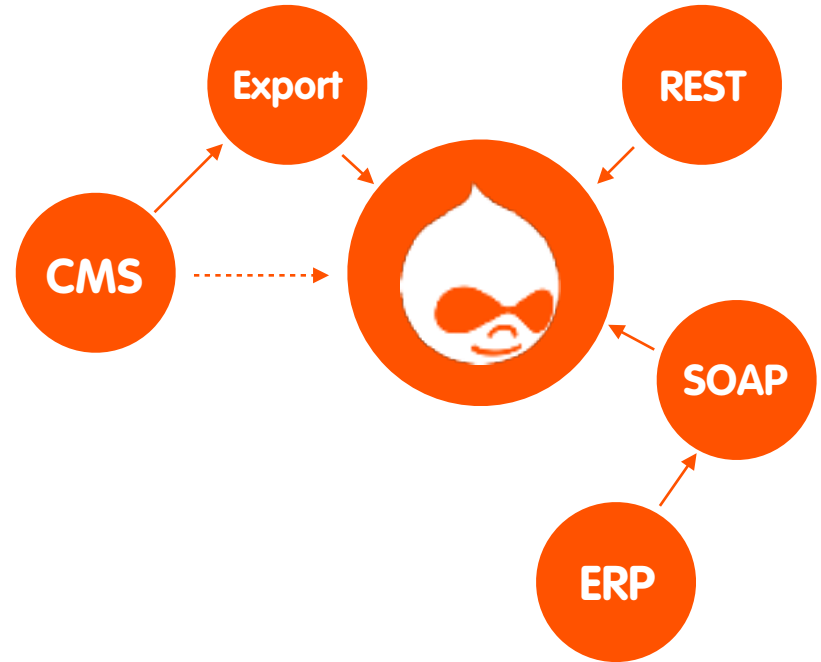
- Restructure data to fit content model
- Enrich data internally or other sources
- Variety of process plugins available
- Custom plugins

Destination

- By default Drupal 8 content entities
- Custom destinations can be
 - external services
 - other sites
 - Drupal 8 core functionalities
 - Custom destination plugins for e.g. posting to REST API

Data sources

Virtually any
structured data source
can be used.



03

Case examples

Case studies

Case study 1: **Product catalogue**

Case study 2: **Social media feeds**

Case study 3: **Outward connections**

Case study 1: Product catalogue

Complex mapping of the business logic vs presentation layer needs

Importer config

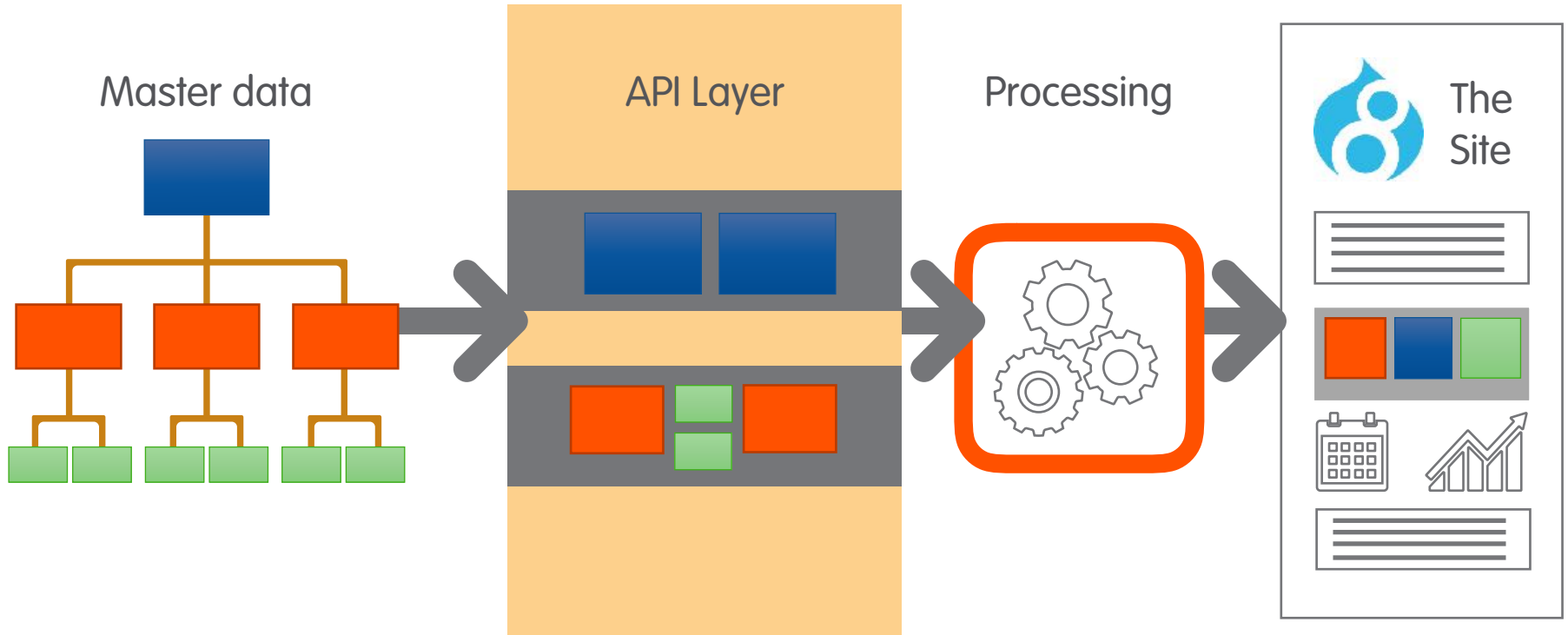
What is the atomic level in master data system and Drupal

Individual prices not show on site, but collections of them

Existing content enriched by new integration

Automatic updating of pricing information from the Master Data System

Case study 1: Product catalogue



Case study 1: Product catalogue

- Based on core and Migrate Plus plugins
- Migration plugins developed earlier were used.
- Querying existing data: migration lookup + custom plugins to query database
 - Example use case: populating the parent of taxonomy referred to by the current item being processed
- **Magic is in the configuration**
 - Config handles the big picture and content mapping between systems / concepts
 - Code handles individual (field level) transformation

Case study 2: Social media feeds

Fetching information from external sources via their APIs

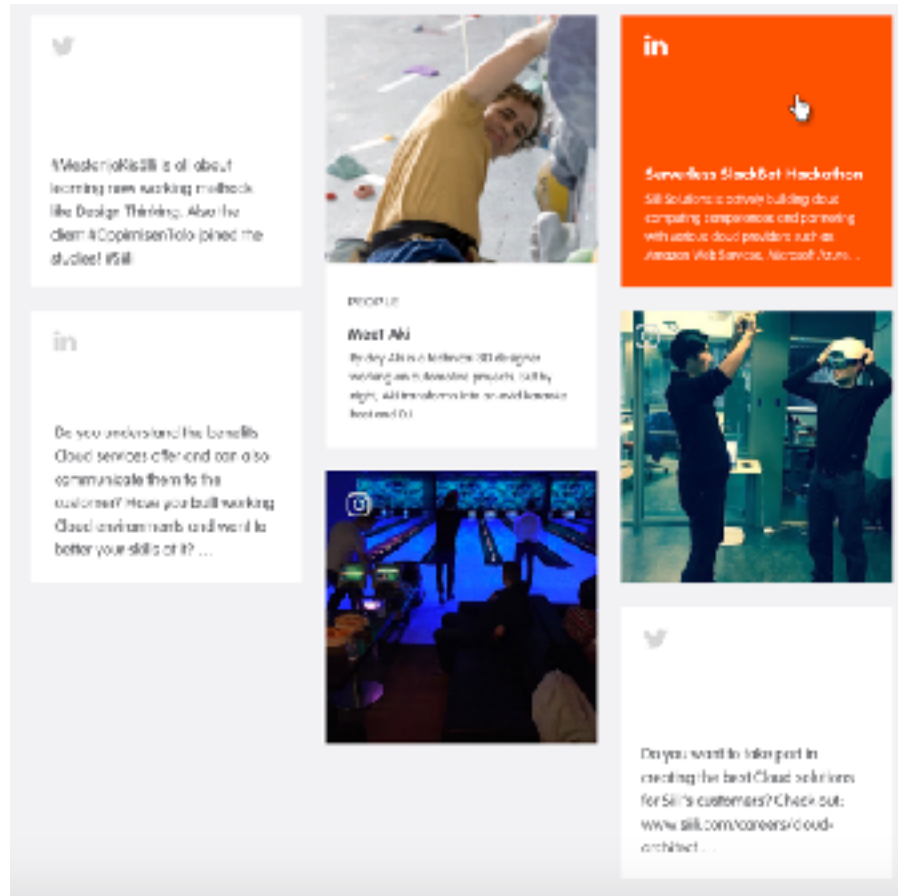
Use of existing libraries to interface with the APIs


Importer config for defining API interaction

Custom entity: social media post


Displaying external content alongside with Drupal content

Case study 2: Social media feeds






WesterloKisill is all about learning new working methods like Design Thinking. Also the client #CopimisenTolo joined the studio! #Sill




Do you understand the benefits Cloud services offer and can also communicate them to the customer? How do you build working Cloud environments and want to better your skills at it? ...



PROFILE


Wael Abi

Wael Abi is a freelance UI designer working as a freelance product designer, helping small businesses into an online business (not just on).



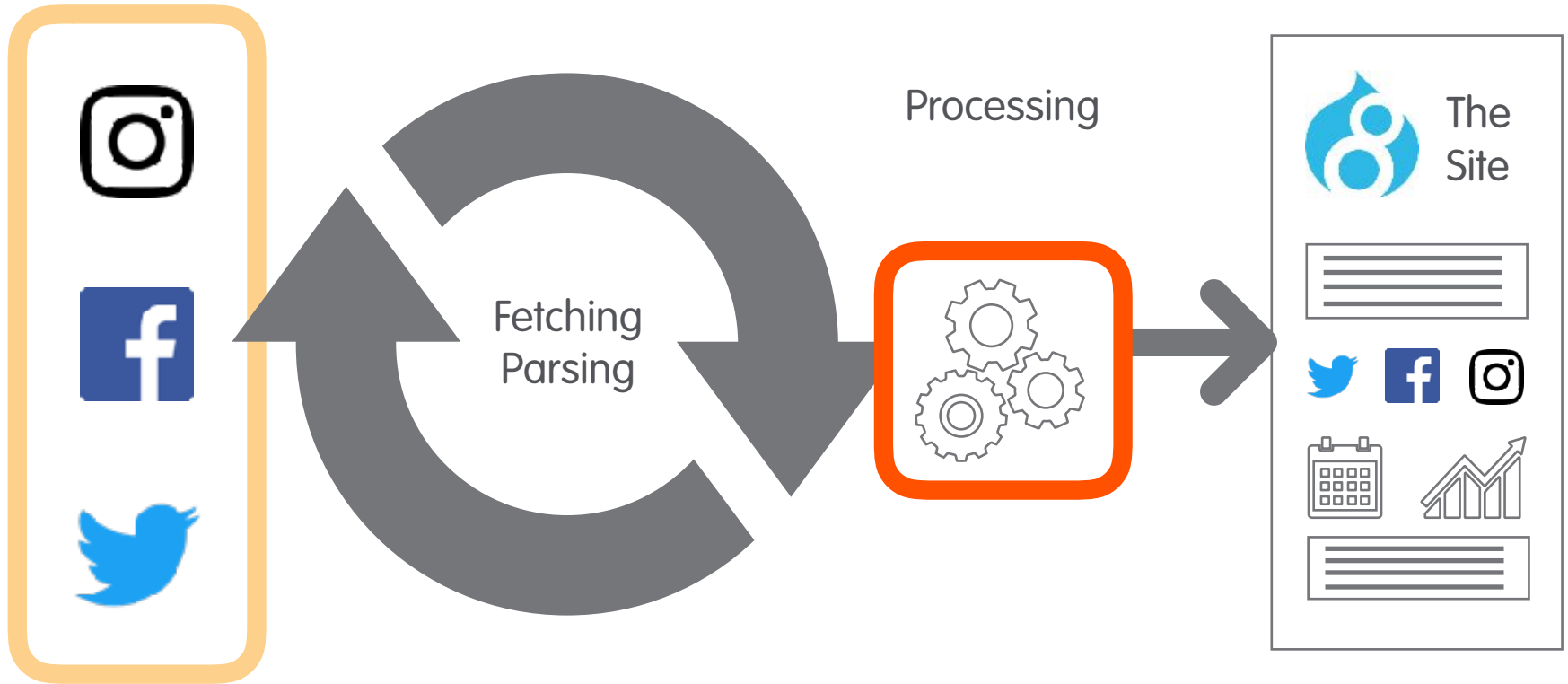
Serverless Stack66t Hackathon

Sill Solutions is always building about computing, containers and packaging with various cloud providers such as Amazon Web Services, Microsoft Azure, ...



Do you want to take part in creating the best Cloud solutions for Sill's customers? Check out: www.sill.com/careers/cloud-architect ...

Case study 2: Social media feeds



Case study 2: Social media feeds

- Follows the regular migration process as overviewed earlier in the presentation
- Source
 - API interaction defined in configuration
 - authentication: OAuth2 custom plugin
- Process
 - Migrate Plus
 - external libraries for API calls
 - Media asset handling
- Destination
 - custom entity types

Case study 2: Social media feeds

- Built to be extensible
- Adding a new social media feed should be simple
- To be open-sourced

Case study 3: Outward connections

Several sites with Drupal and other CMSs

Content needs to be movable between systems

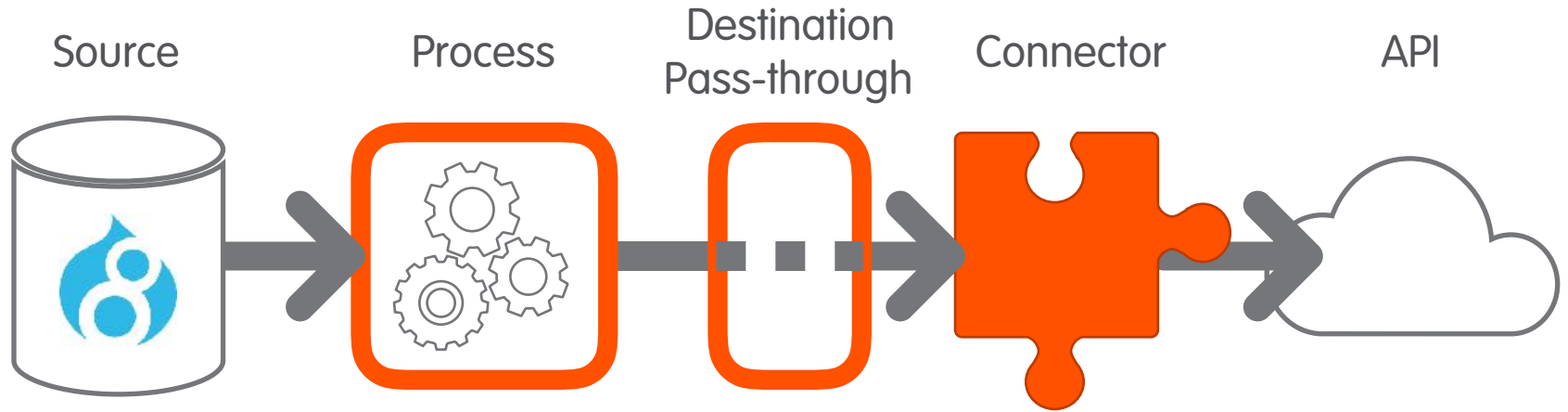
APIs available on all systems

Functionality must be extendable and modular

Case study 3: Outward connections

- A custom source plugin based on Drupal-to-Drupal migrations
- Content mapping in process phase
- Destination passes on the mapped data as objects
- A service for posting data to APIs

Case study 3: Outward connections



04

Solution considerations

Working with API endpoints

- Different endpoints provide datasets of different aspects of the whole data
- Combinations of responses from different endpoints make up the total presentable information

Data model and presentation

- Master data has its own data structure that serves the information value
- Drupal's data structure (content model) is built to serve presentation layer
- Migration pipeline functions as the mapping between atomic structure of master data system and desired presentation structure
- Separate metadata as taxonomies

Leveraging existing solutions

- In social media importer we could utilise existing PHP SDK libraries for different social media services
- PHP has good support for all different structured formats.
- Existing Drupal migration plugins should be extended instead of writing the whole implementation from scratch.
- Write once, use many times

Customize Drupal to your needs

- Drupal 8 supports building content entity types on top of base types, not everything is a node anymore.
- Custom entity types are easy for any developer to implement.
- Plan and design content structure in Drupal to balance with keeping structural significance while making the content simple enough to be manageable.
- Developers need to be aware of the reasoning behind the structure, relations and classifications to be able to model content correctly. Help them understand the data source.

Integration concerns

- When connecting separate systems, there needs to be expertise from both sides.
- Make sure the API supports all use cases of the integration will be applied to.
 - If the API is under development, get the migration developers into contact with API developers.
- Define specifications for change management
 - How to handle deleting content
- Integrating with existing content can require manual work.

siili_

Thank you!

Follow us on Twitter

@SiiliSolutions

@jnous

@narnua

Become a Drupal contributor

Friday from 9am

- First timers workshop
- Mentored contribution
- General contribution