

SULU IN 2025

A Year of success

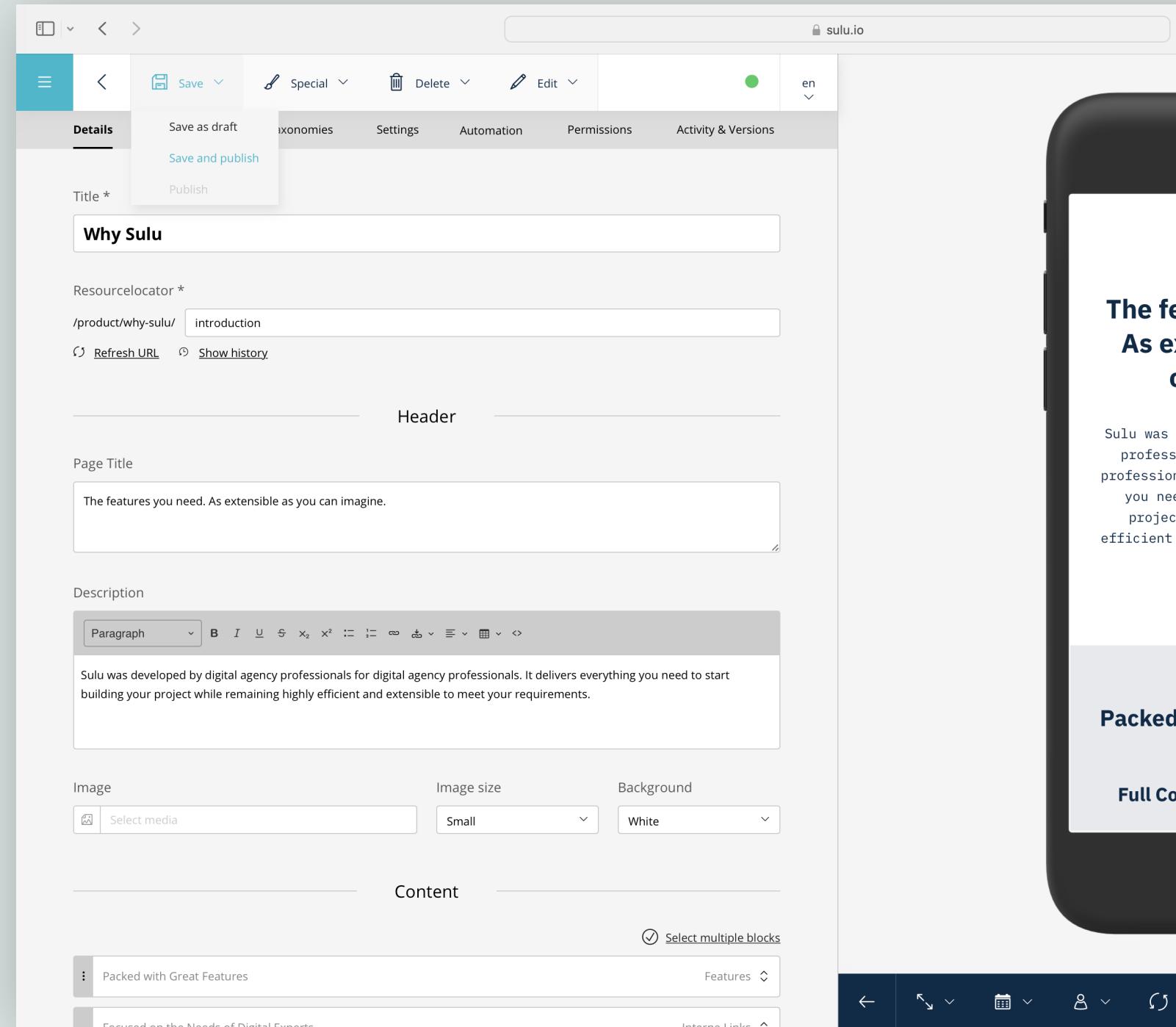
Hi, I'm Johannes

- Core Developer at Sulu CMS & Creator of various open-source stuff
- Works with Symfony since 2012
- Lecturer at University of Applied Science Dornbirn
- Open Source & AI Enthusiast
- Father of two



Open source. Fit for enterprise.

- Enterprise content management platform
- Built full-stack on the Symfony framework
- Made for businesses
- Intuitive UI with great UX
- High performance
- 100% Open Source



The screenshot shows the Sulu CMS interface for editing a page titled "Why Sulu". The top navigation bar includes "Save", "Special", "Delete", "Edit", and language selection ("en"). Below the title, there's a "Resource locator" field containing "/product/why-sulu/introduction". The main content area has sections for "Header", "Page Title", and "Description". The "Page Title" section contains the text "The features you need. As extensible as you can imagine.". The "Description" section contains the text "Sulu was developed by digital agency professionals for digital agency professionals. It delivers everything you need to start building your project while remaining highly efficient and extensible to meet your requirements." At the bottom, there are sections for "Image", "Image size", "Background", and "Content". The "Content" section includes a "Select multiple blocks" checkbox and two blocks: "Packed with Great Features" and "Focused on the Needs of Digital Experts". A sidebar on the right lists "The features you need. As extensible as you can imagine.", "As extensible as you can imagine.", and "Focused on the Needs of Digital Experts".

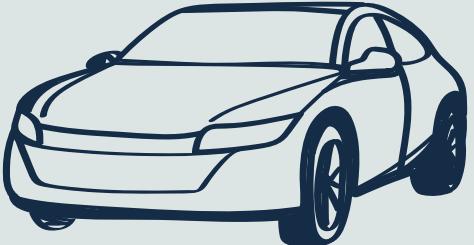


Made for heavy loads



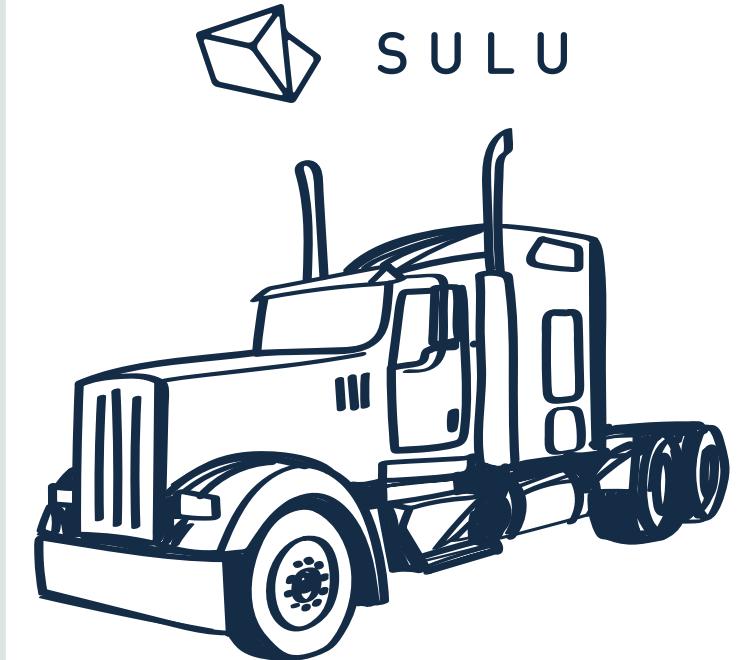
Bicycles

Everyone can ride them,
many can repair it.



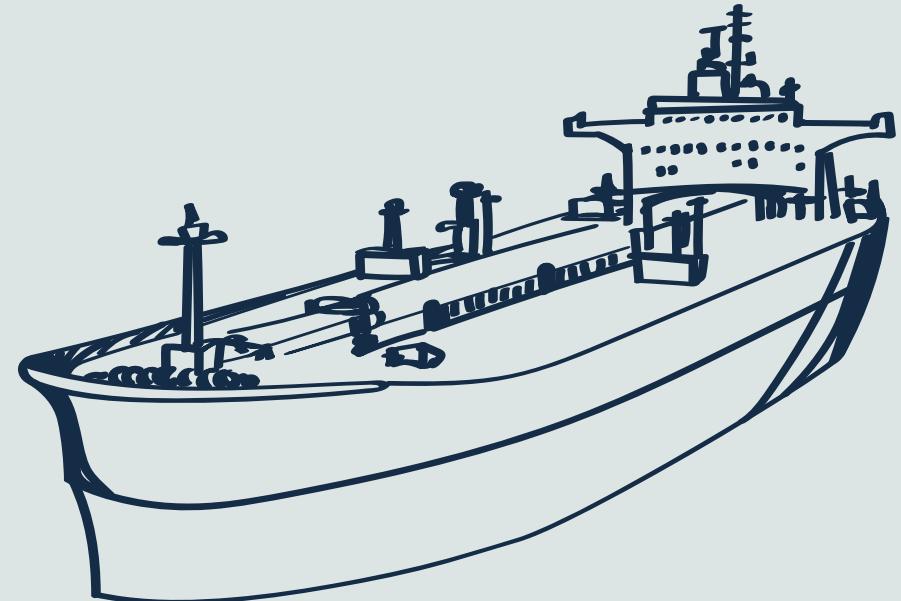
Cars

Many can ride them,
some can repair it.



Trucks

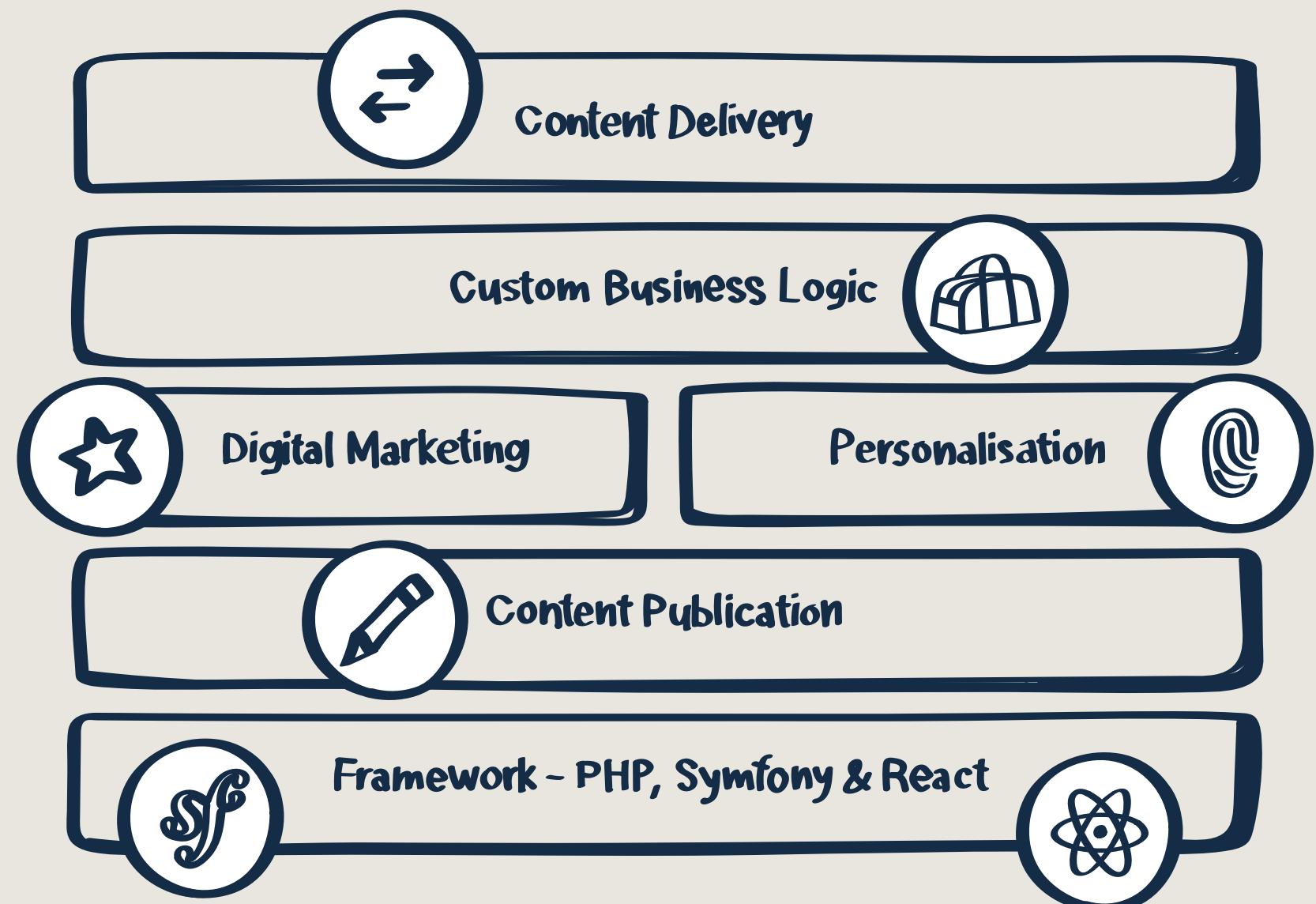
Need a special driving-
license, must be configured
to your needs.



Supertanker

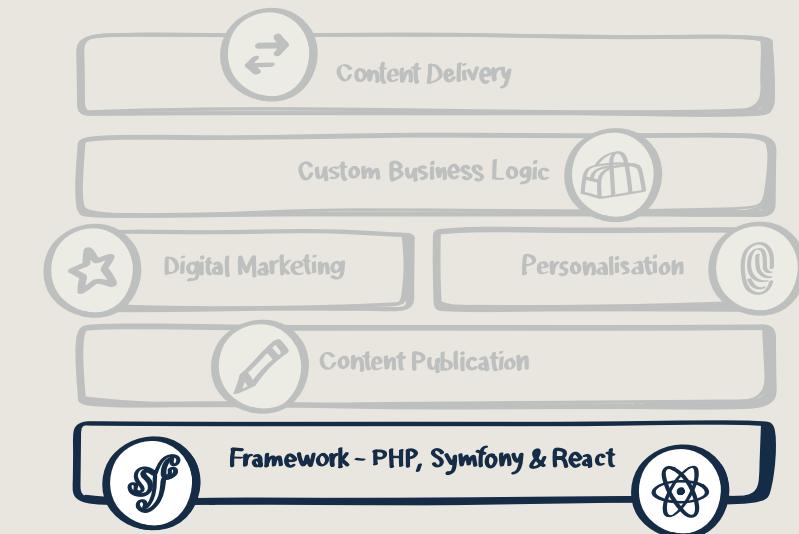
Need highly specialized staff,
expensive and very complex.

Building Blocks



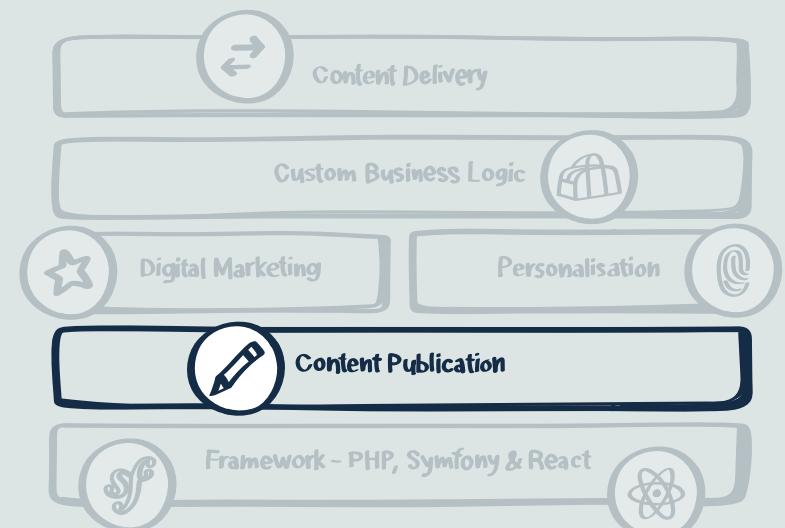
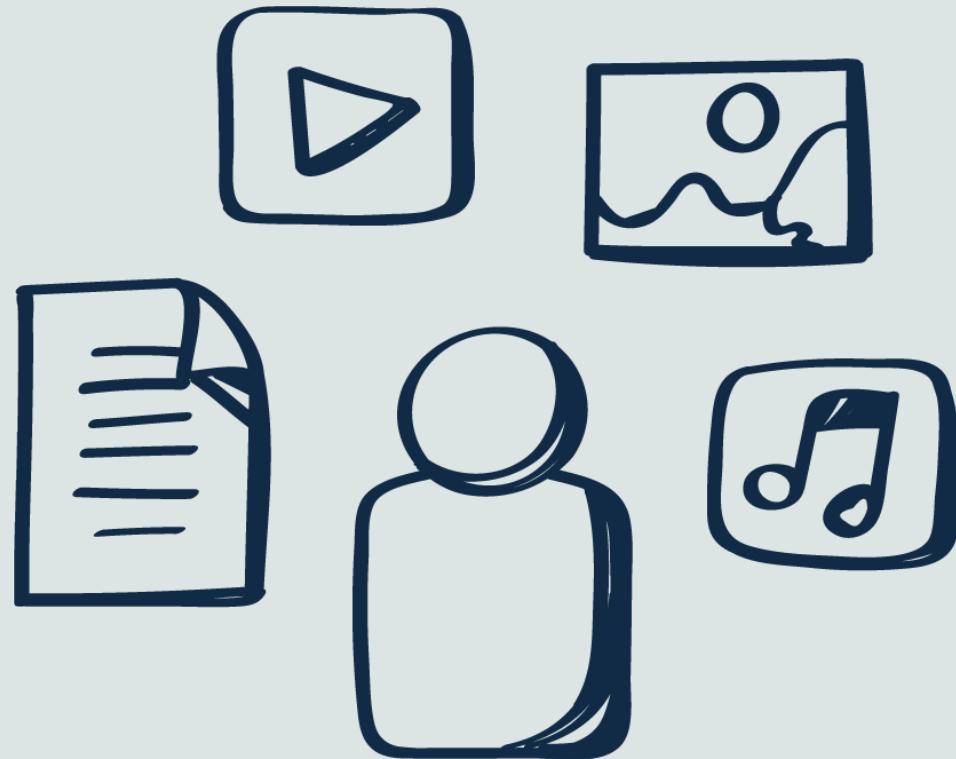
Sulu is frame-work based

- Don't reinvent the wheel
- Built on top of a solid foundation
- Re-using proven and widely tested software design patterns
- Secure and long term maintained
- Full-Stack Symfony Framework
- More than 13 billion downloads



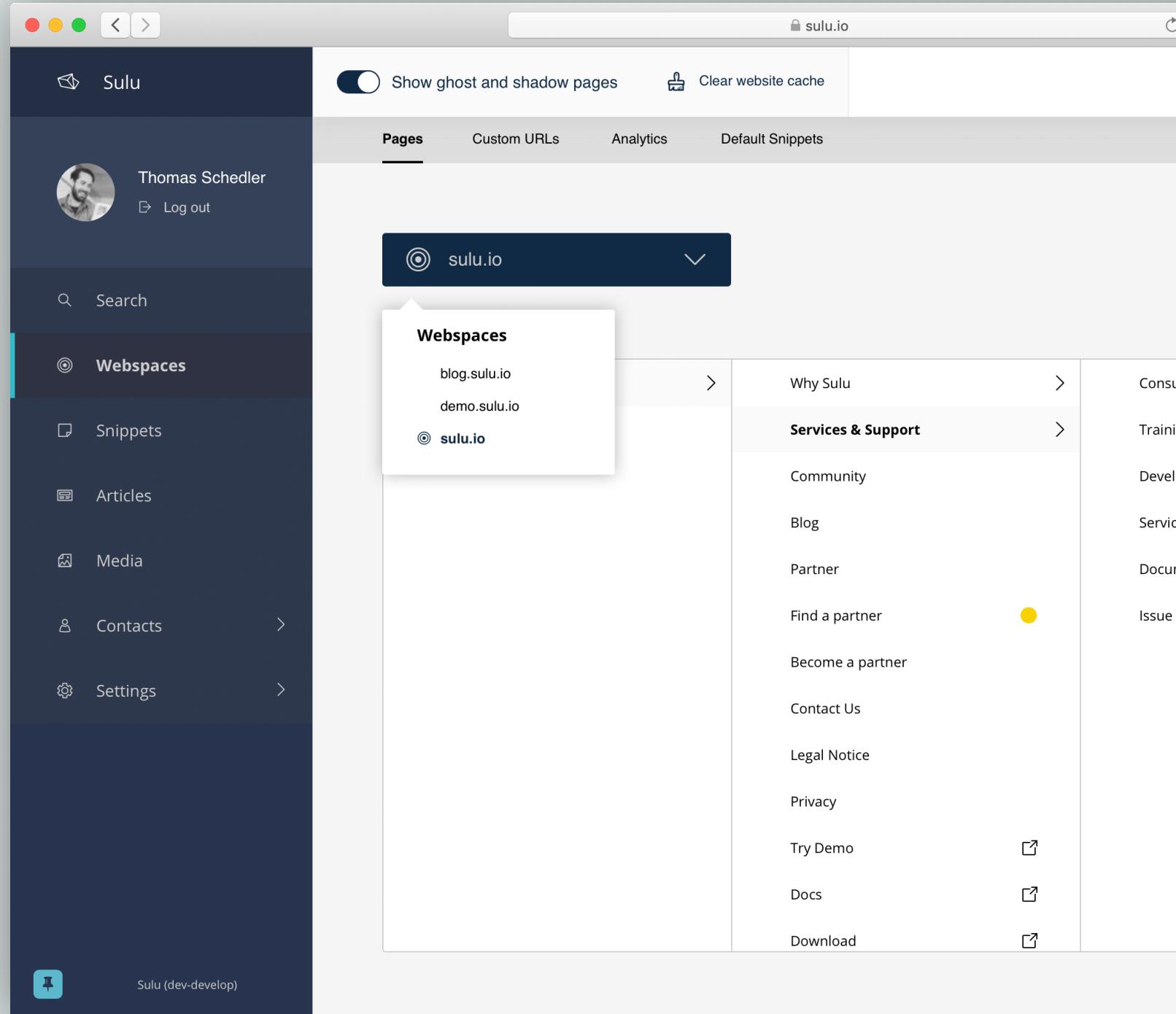
Great for content publication

- Full content life cycle -> create, edit, draft and publish content
- Consistent & intuitive UI
- Live preview
- Versioning
- Multiple platforms managed in one place



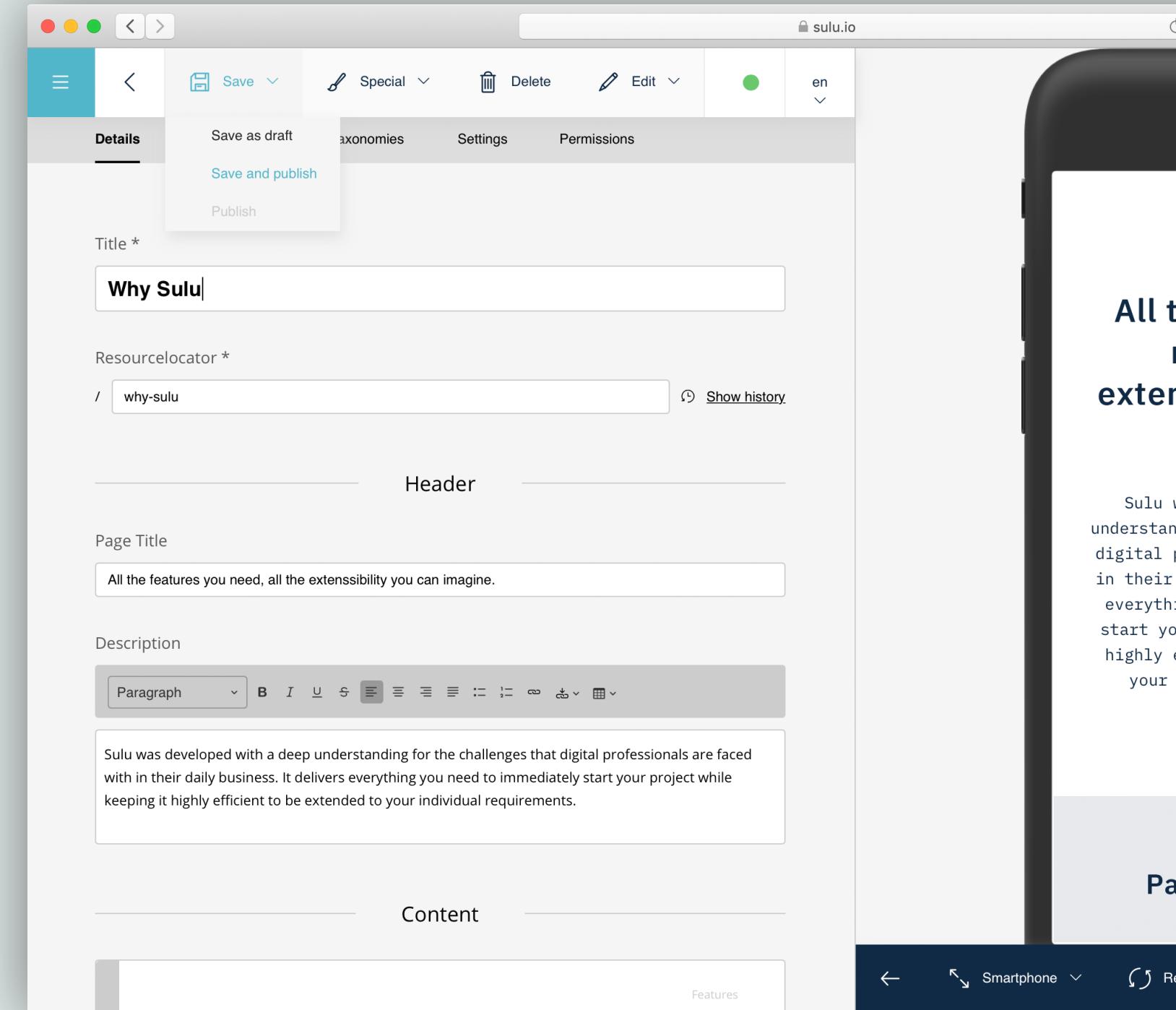
Webspaces

- One single content-structure (called page tree)
- The structure represents one or more websites
- Multiple languages implemented as dimensions
- Multiple webspaces support



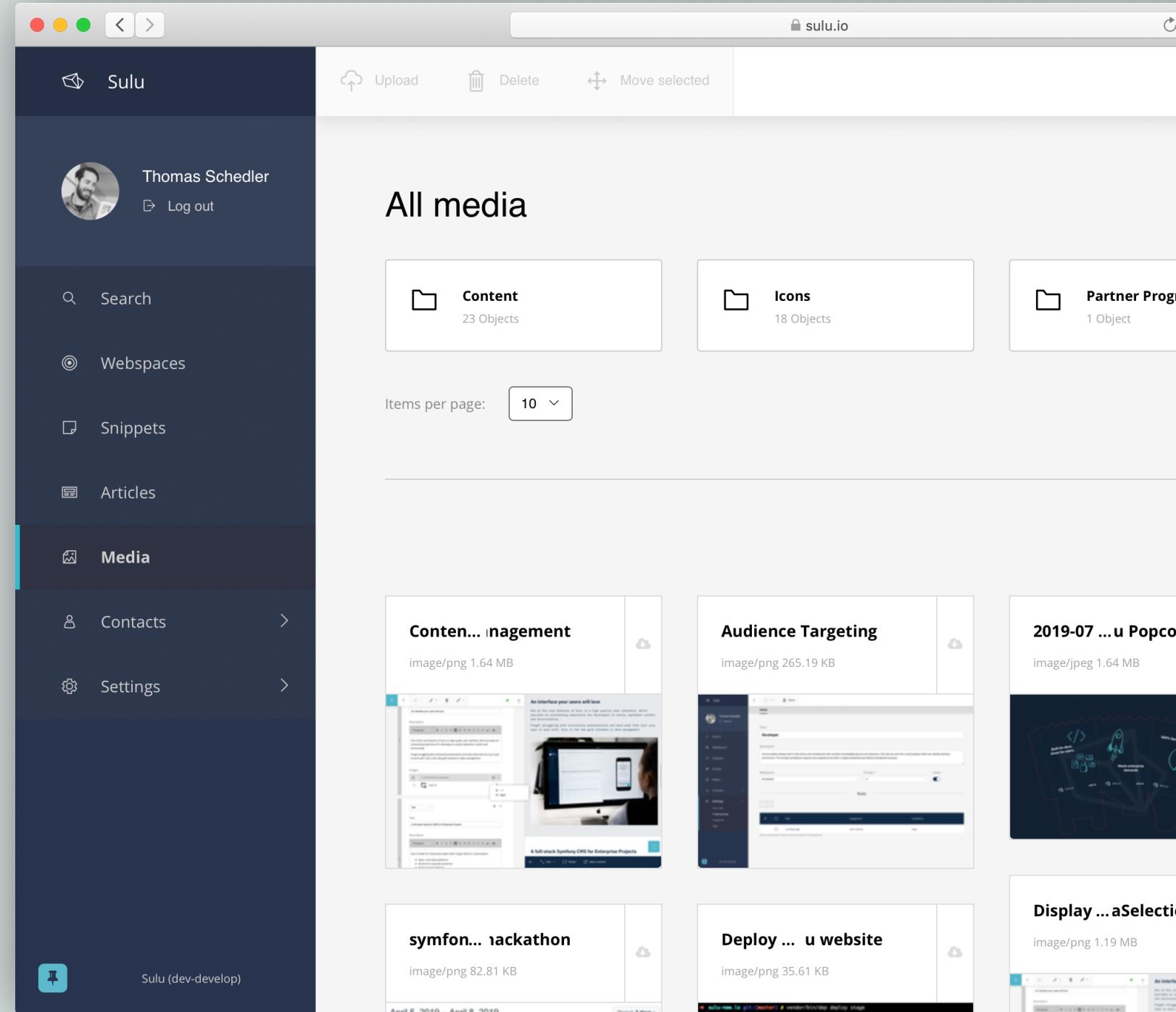
Templating

- Powerful template engine
- Each page template is defined by:
 - an XML file that contains the page structure
 - a Twig file that contains the HTML representation
- The page structure consists of properties, each of which has a content type



Media Management

- Centralized repository for documents, images etc.
- Independent management based on folder hierarchies
- Multi-language support for document metadata
- Document versioning
- Automated image re-sizing & optimization



The Two Pillars: Technology and Usability

- **Technology:**
 - Built on modern frameworks like Symfony for flexibility and power
 - Designed to scale with evolving web needs (e.g., APIs, headless CMS, AI)
- **Usability:**
 - A CMS that editors love using
 - Reducing complexity while keeping flexibility



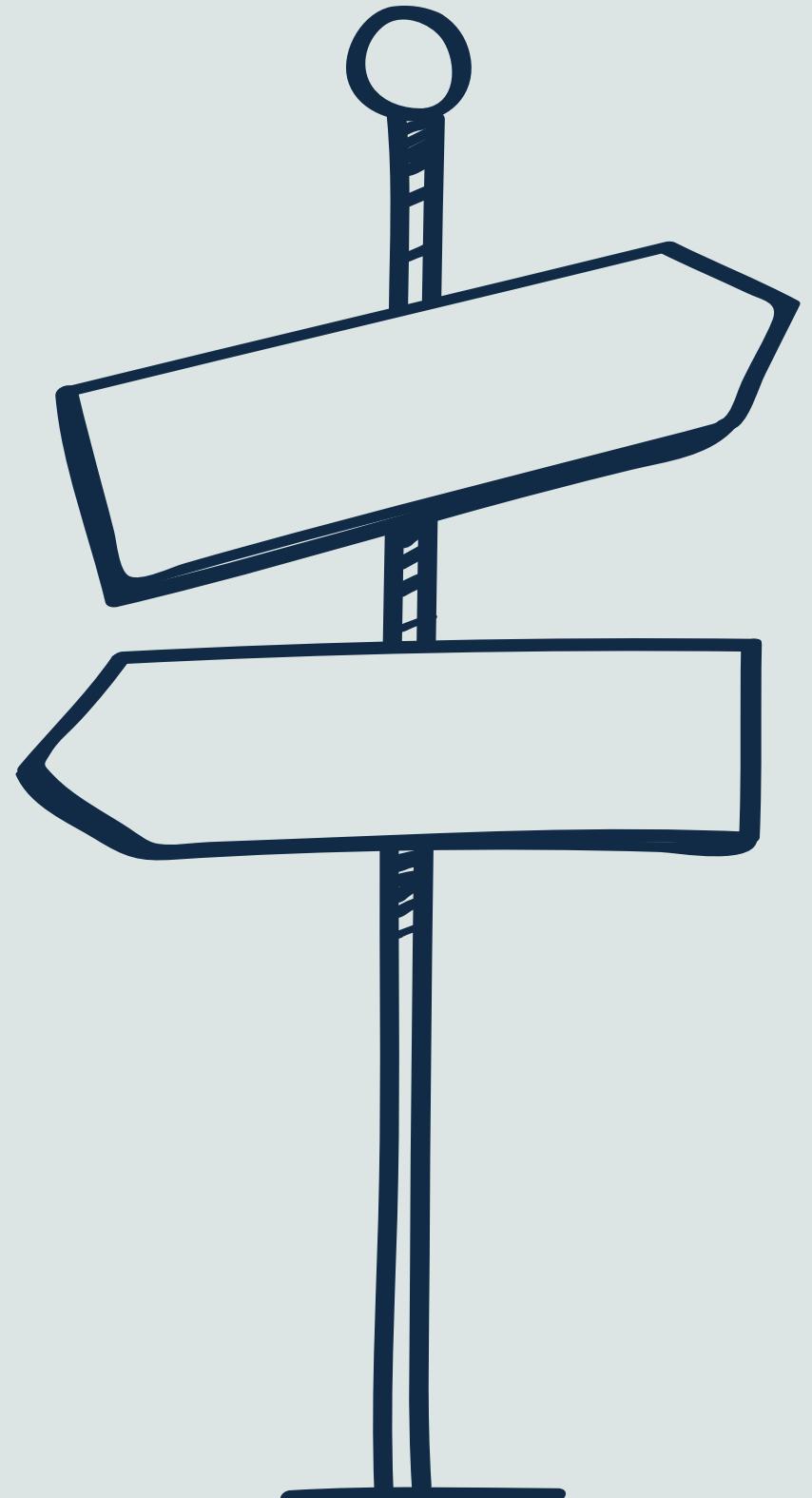
“ ”

*A successful CMS must serve
both **developers** and **content
editors** equally.*



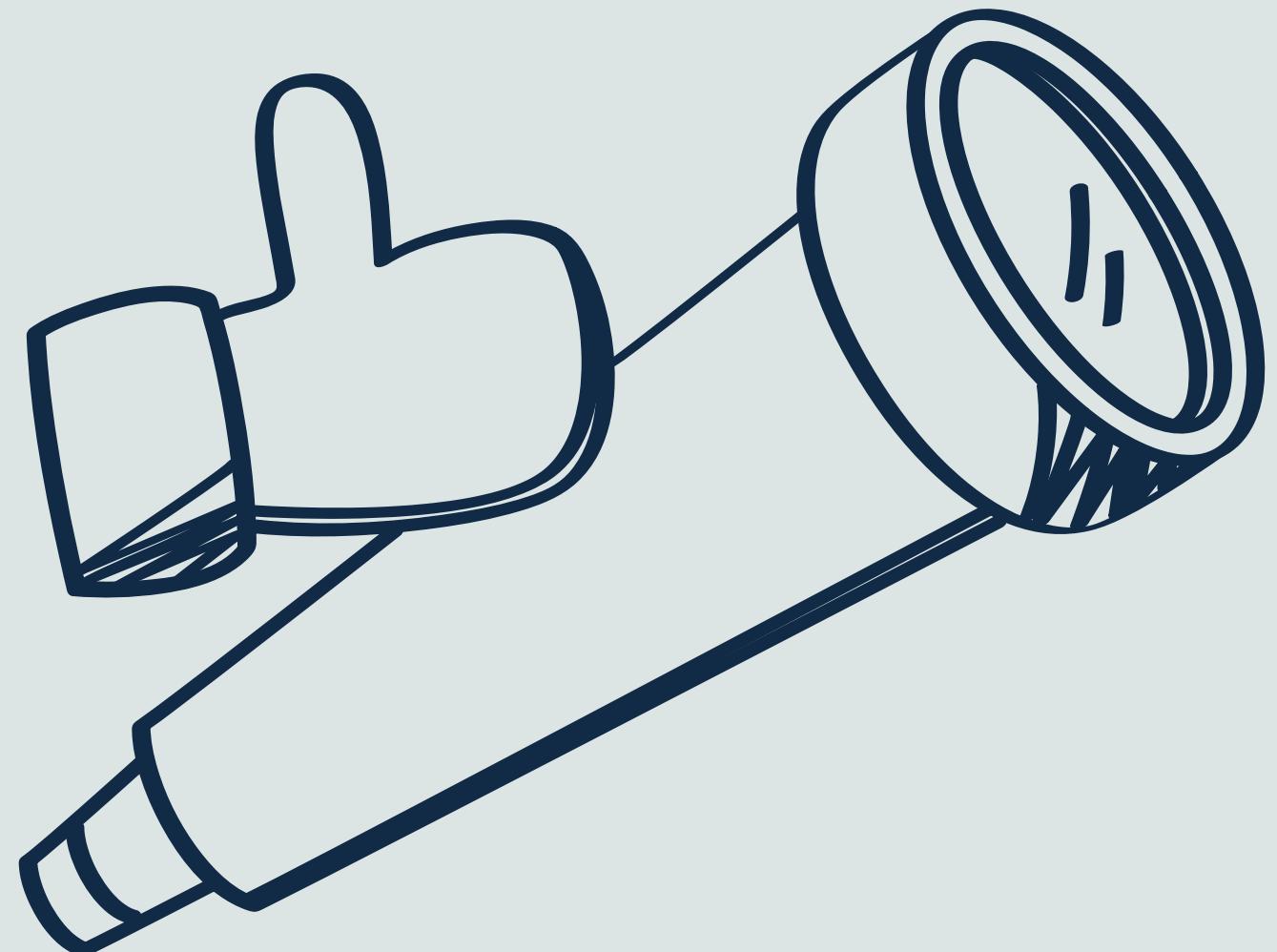
What We've Learned Along the Way

- **Content Management:** It's often underrated but remains critical for long-term success
- **Learn to Say No:** Protecting the project's vision requires making tough decisions
- **PHP's Strength:** Despite skepticism, PHP continues to be a reliable and evolving technology
- **Balancing Act:** Finding the balance between innovation and business needs is an ongoing challenge



The Future of Sulu: Continuing the Legacy

- **Vision:** Empowering teams with usability-driven technology
 - Sulu 3.0 and the advanced Content Storage system
 - AI integration: Streamlining workflows for smarter content management
- The best is yet to come!



THE FUTURE OF SULU

Continuing the Legacy



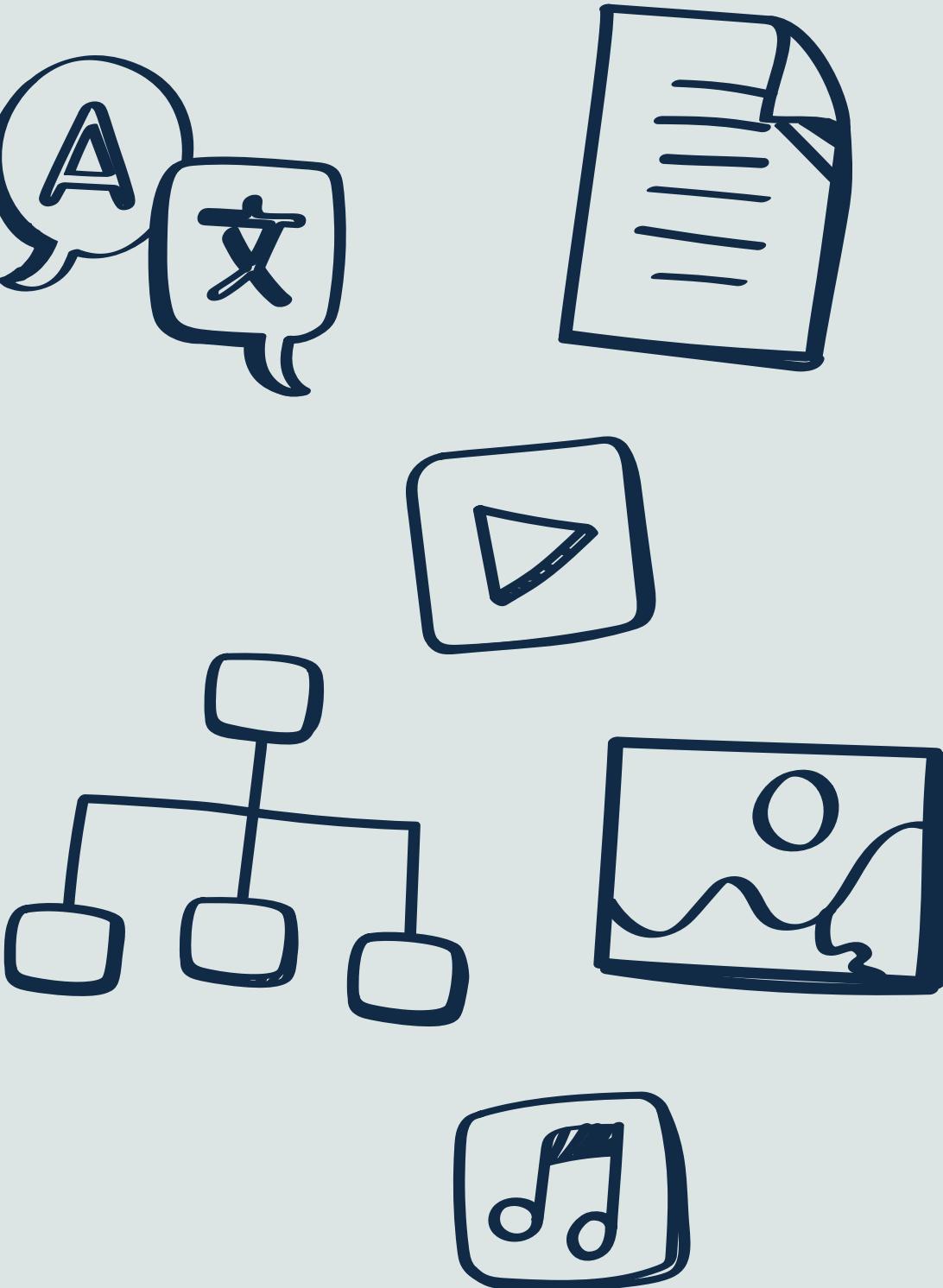
A New Era with Sulu 3.0

- Sulu 3.0 marks the next big chapter in our journey
- This release has been long in the making, but we finally see the light at the end of the tunnel
- Replacing the content storage was a monumental operation – now it's ready to redefine how we manage content



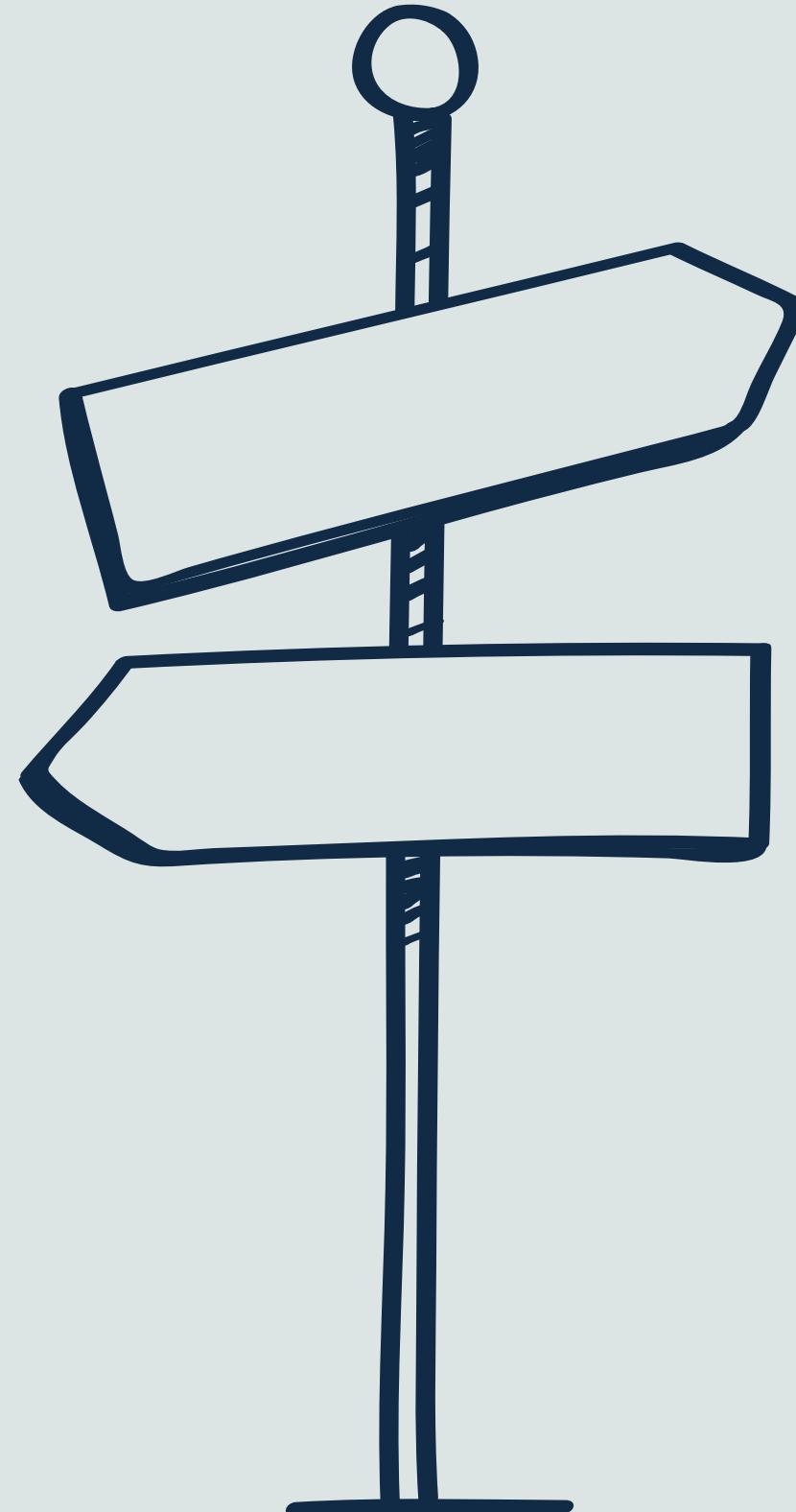
Why Storing Content Is Hard

- CMS content is **unstructured** and **hierarchical** – it doesn't fit neatly into traditional tables
- Relational databases struggle with complex hierarchies and dynamic schemas
- Content storage needs to evolve with the needs of modern CMS



Transition from PHPCR

- We started with PHPCR, inspired by the Java Content Repository (JCR)
- PHPCR allowed us to build quickly without creating storage from scratch
- Today, however, we need a solution that is faster, more straightforward, and seamlessly integrated with the PHP and Symfony ecosystem



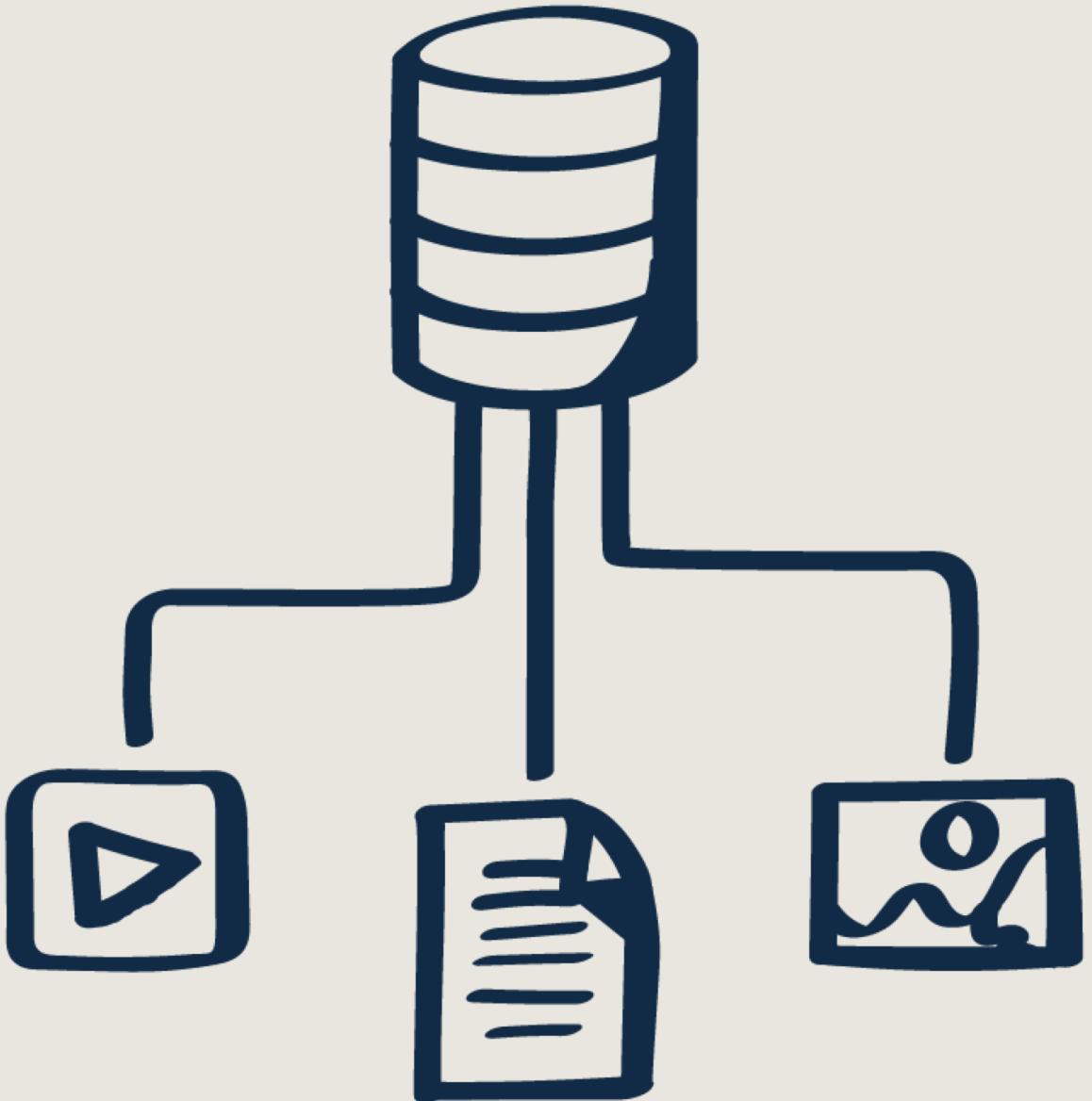
Introducing the New Content Storage

- Sulu 3.0 introduces a new content storage solution, built from the ground up for performance and flexibility
- Implemented with **Symfony / Doctrine** and optimized for semi-structured data
- Already running in production for some time, proving its reliability and performance in real-world scenarios

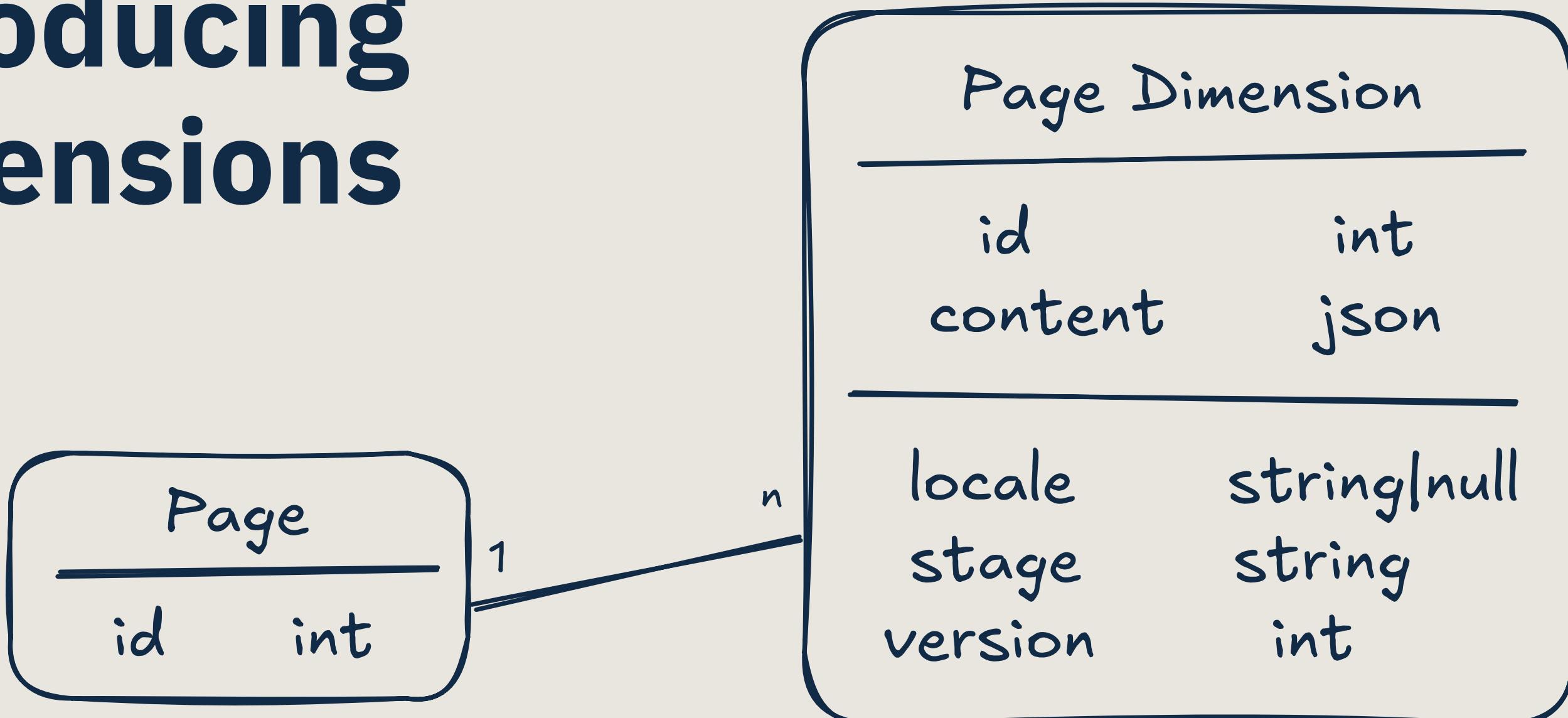


Requirements

- **Optimized Loading:** Fetch only the content and languages you need – no overfetching
- **Built-in Features:** No additional services required for versioning, drafting, and other essential capabilities
- **Flexible Architecture:** Pages and Articles are simply Doctrine ORM entities
- **Content-Ready for Entities:** Any entity can be enhanced to support content
- **Powerful Queries:** Access content seamlessly using SQL or DQL queries



Introducing Dimensions





Page Dimension

id	locale	stage	version	content
1	de	edit	0	{"description": "..."}
2	en	edit	0	{"description": "..."}
3	NULL	edit	0	{"key": "..."}
4	de	version	1726605235	{"description": "..."}
5	en	version	1726605235	{"description": "..."}
6	NULL	version	1726605235	{"key": "..."}
7	de	live	0	{"description": "..."}
8	en	live	0	{"description": "..."}
9	NULL	live	0	{"key": "..."}



Page Dimension

id	locale	stage	version	content
1	de	edit	0	{"description": "..."}
2	en	edit	0	{"description": "..."}
3	NULL	edit	0	{"key": "..."}
4	de	version	1726605235	{"description": "..."}
5	en	version	1726605235	{"description": "..."}
6	NULL	version	1726605235	{"key": "..."}
7	de	live	0	{"description": "..."}
8	en	live	0	{"description": "..."}
9	NULL	live	0	{"key": "..."}



Page Dimension

id	locale	stage	version	content
1	de	edit	0	{"description": "..."}
2	en	edit	0	{"description": "..."}
3	NULL	edit	0	{"key": "..."}
4	de	version	1726605235	{"description": "..."}
5	en	version	1726605235	{"description": "..."}
6	NULL	version	1726605235	{"key": "..."}
7	de	live	0	{"description": "..."}
8	en	live	0	{"description": "..."}
9	NULL	live	0	{"key": "..."}

Merging Dimensions

Page Dimension

id	locale	stage	version	content
7	de	live	0	{"description": "..."}
9	NULL	live	0	{"key": "..."}



	de	live	0	{"key": "...", "description": "..."}
--	----	------	---	--------------------------------------

Consistent Public Interface, Evolving Under the Hood

JSON API Response

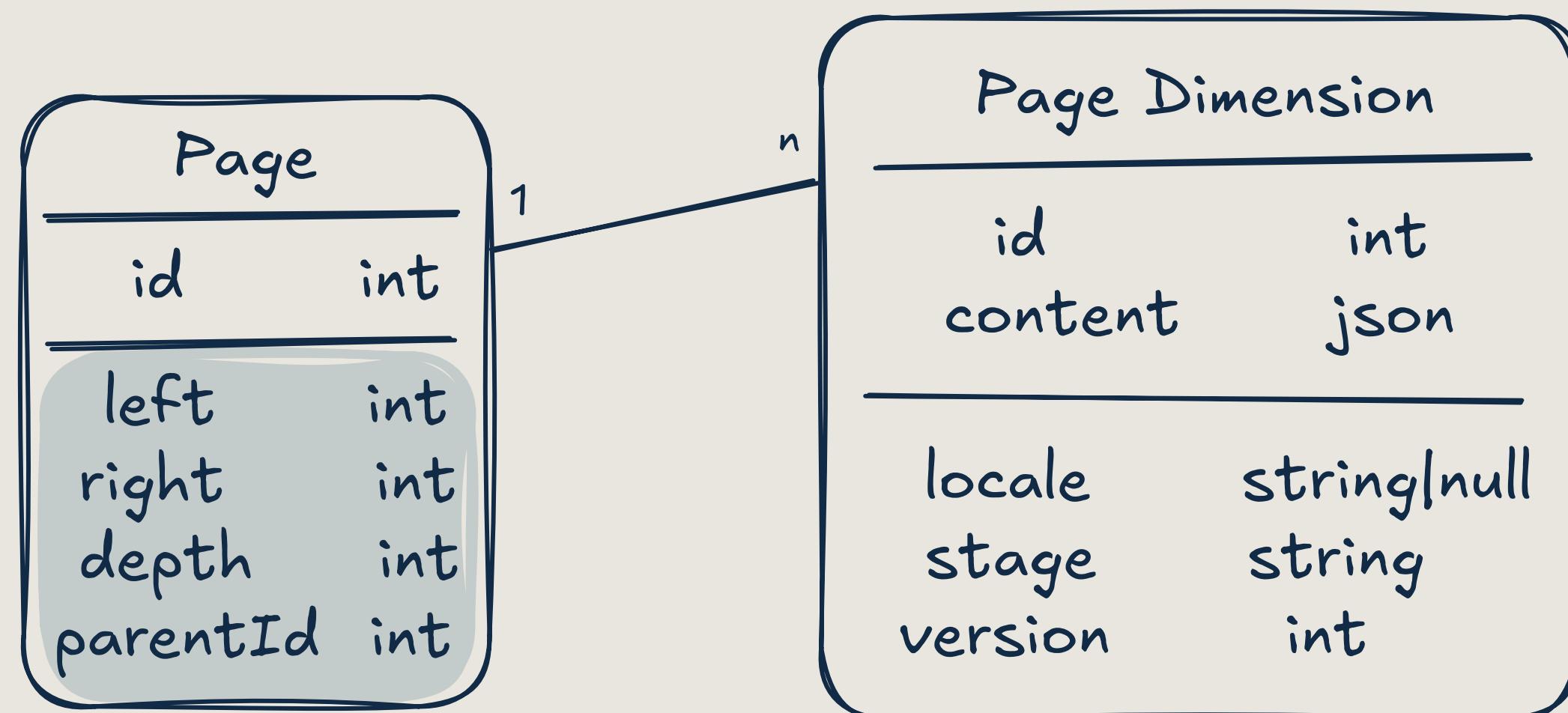
```
{  
  "id": 1,  
  "locale": "de",  
  "key": "...",  
  "description": "..."  
}
```

TWIG Rendering Content

```
<div id="{{ content.key }}>  
  {{ content.description|raw }}  
</div>
```

Adding Database Columns to Optimize Query Performance

Enhance the Main Entity with Structure Related Information



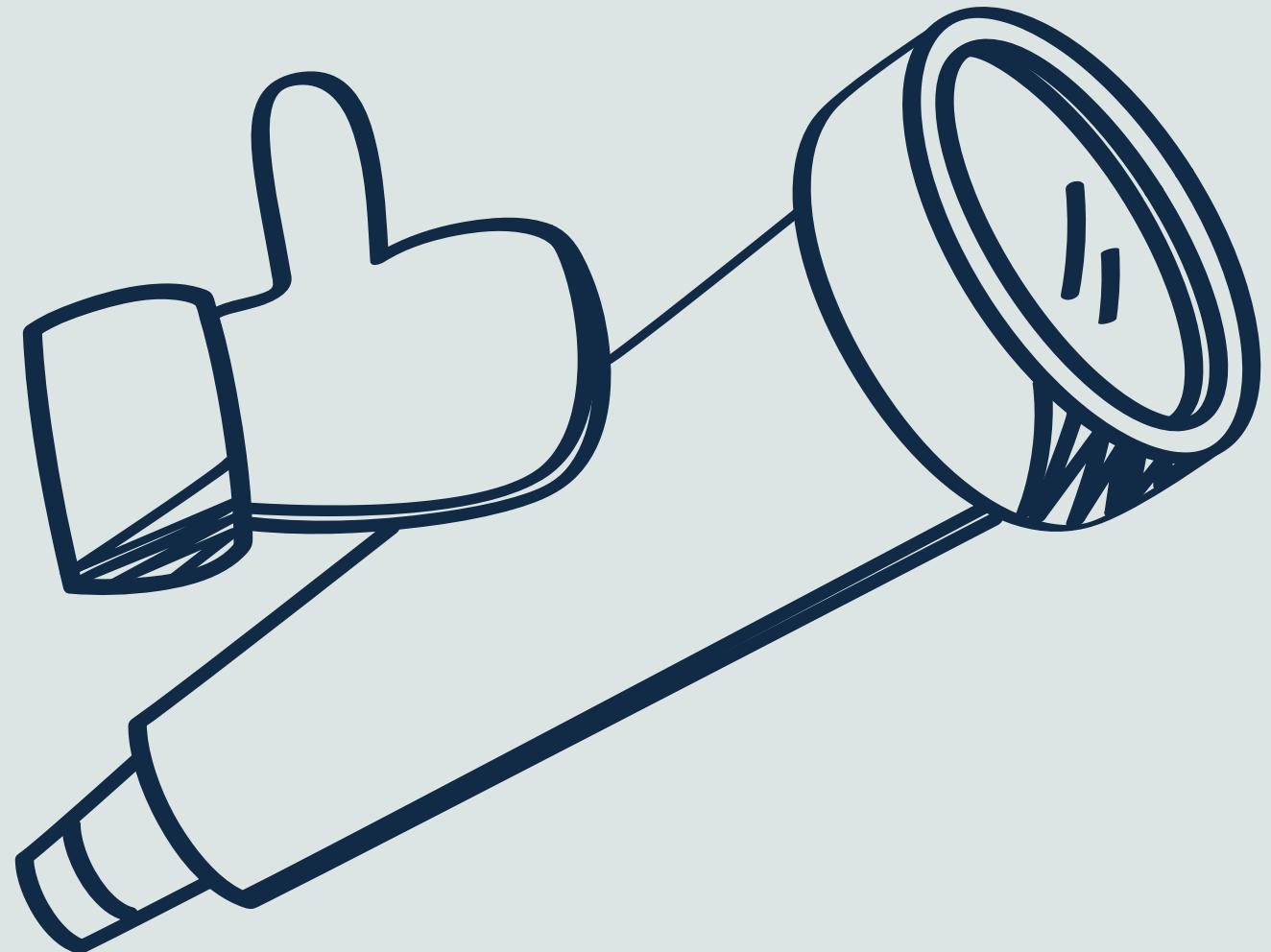
Migration Path for Existing Content

- A smooth migration path makes switching from PHPCR straightforward
- Tools, documentation, and guides ensure you can adapt without hassle



A Platform for the Future

- This is just the beginning – our vision is to continually innovate and optimize
- Alongside the new content storage, we're undertaking significant cleanups across the platform
- Dependencies are being updated, including pumping up versions of Symfony, PHP, and others, to stay modern and secure



OPTIMIZING CONTENT WITH AI

Sulu's Advanced AI Toolkit and Service

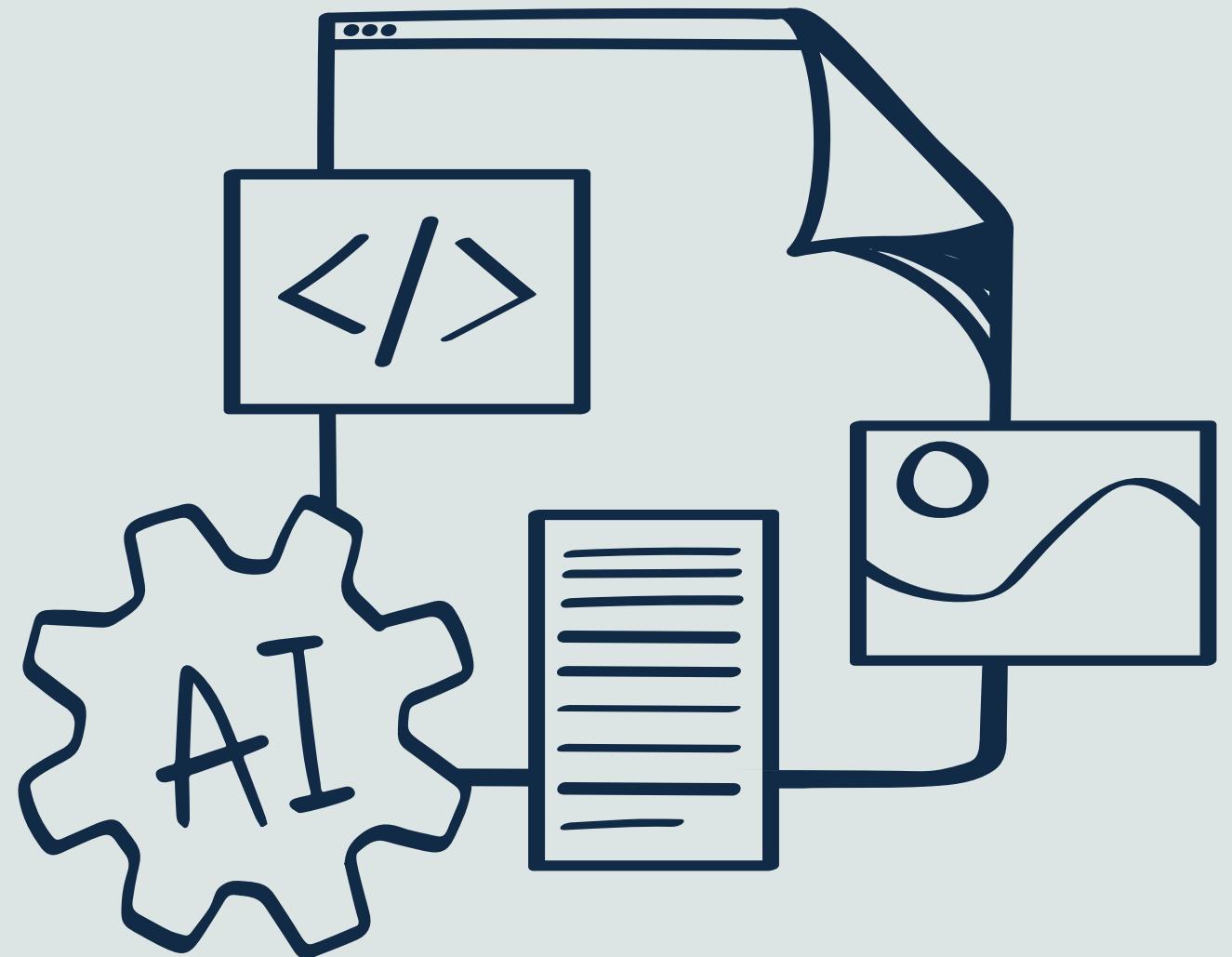
AI is Here to Stay – Adopt and Leverage

- AI is no longer a trend – it's a permanent part of technology
- Companies must embrace AI or risk falling behind
- The question isn't *if*, but *how* to leverage AI



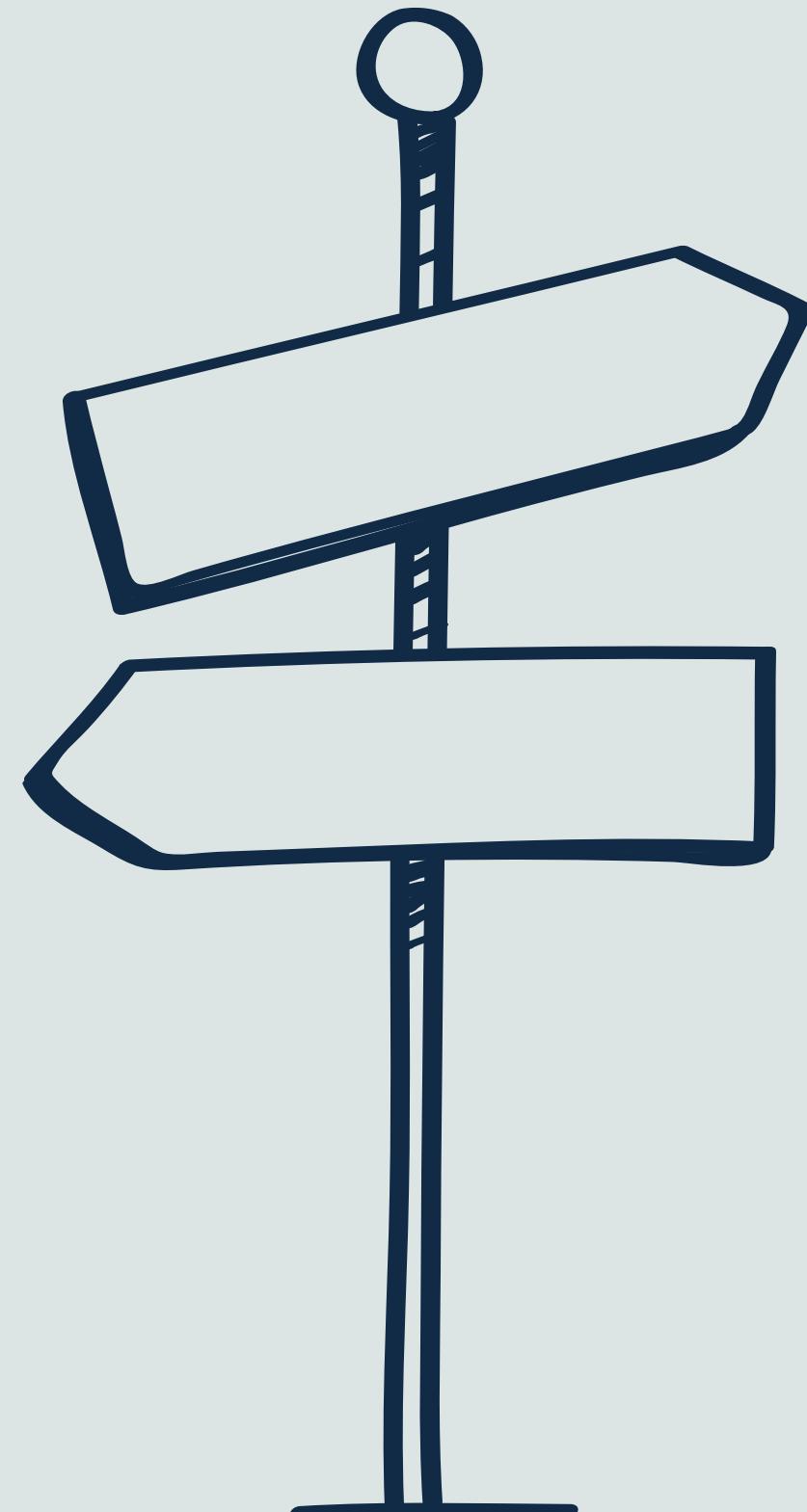
Why AI Matters for Content Production

- Content production challenges:
 - SEO, translations, user engagement, accessibility, output channels
- AI/LLMs enables:
 - Automates time-consuming standard tasks like SEO optimization, translations, and accessibility
 - Acts as a creative partner for brainstorming and refining ideas



What is Sulu's Approach to AI?

- Empowerment Over Replacement:
 - AI amplifies, not replaces, your team's efforts
- Integrated AI
 - AI is seamlessly embedded in the CMS, minimizing disruption
 - Make it easier for developers and agencies to build customized AI solutions



The Sulu AI Bundle – Open Source at Its Core

- Abstraction layer for multiple AI providers
- Advanced field-based operations like translation and optimization
- Ensures transparency and flexibility.
- Sulu stays true to its roots while leading innovation.
- Balancing innovation with accessibility in open source.



ANTHROPIC

Introducing Sulu AI Platform – Our First Commercial Product

- Advanced AI services are often commercial and costly.
- Centralized AI-driven content optimization service
- Integrated multi-provider support
- Tailored towards the needs of agencies



MODELFLOW-AI

Open Source Framework for Sulu's AI Integration

Modelflow-AI

- One Interface, Multiple Providers
- Criteria-based Model Selection
- Flexible Integration
- Feature Rich (Streaming, Multi-Modal, Image Generation, ...)



Modelflow Code Slides

- One Interface, Multiple Providers
- Criteria-based Model Selection
- Flexible Integration
- Feature Rich (Streaming, Multi-Modal, Image Generation, ...)

```
<?php

use ModelflowAi\Chat\AIChatRequestHandlerInterface;
use ModelflowAi\Integration\Symfony\Criteria\ModelCriteria;

class ContentGenerator
{
    public function __construct(
        private AIChatRequestHandlerInterface $handler,
    ) {}

    public function generate(string $text): string
    {
        $response = $this->handler->createRequest()
            ->addUserMessage('Generate Text')
            ->addCriteria(ModelCriteria::GPT40)
            ->execute();

        return $response->getMessage()->content;
    }
}
```



```
protected function execute(InputInterface $input, OutputInterface $output): int
{
    $io = new SymfonyStyle($input, $output);

    $messages = [];

    while(true) {
        $question = $io->ask('You');
        if('exit' === $question) {
            break;
        }

        $response = $this->chatRequestHandler
            ->createStreamedRequest(...$messages)
            ->addUserMessage($question)
            ->addCriteria(ModelCriteria::LLAMA3_2)
            ->execute();

        foreach ($response->getMessageStream() as $message) {
            $io->write($message->content);
        }

        $io->newLine(2);

        $messages = $response->getRequest()->getMessages();
        $messages[] = new AIChatMessage(AIChatMessageRoleEnum::ASSISTANT, $response->getMessage()->content);
    }

    return Command::SUCCESS;
}
```

Because Open Source Matters

- composer require modelflow-ai/*
- <https://github.com/modelflow-ai>



SULU.IO

Thank you!