

Migrating Articles from Salesforce to AEM Guides: A Seamless Approach

Enhancing Efficiency, Data Integrity, and Scalability in Content Management

Disclaimer

The views and opinions expressed in this presentation are my own and do not represent the views or official position of my current and previous employers.

The content is based on general industry knowledge and publicly available information. No proprietary or confidential information will be shared during this talk.



Bhanu Phanindra Babu

Background

I am a seasoned technology professional with over 10 years of experience in application development across various industries.

Academically, I hold a Master's degree in Computer Science from the University of Central Missouri, USA.

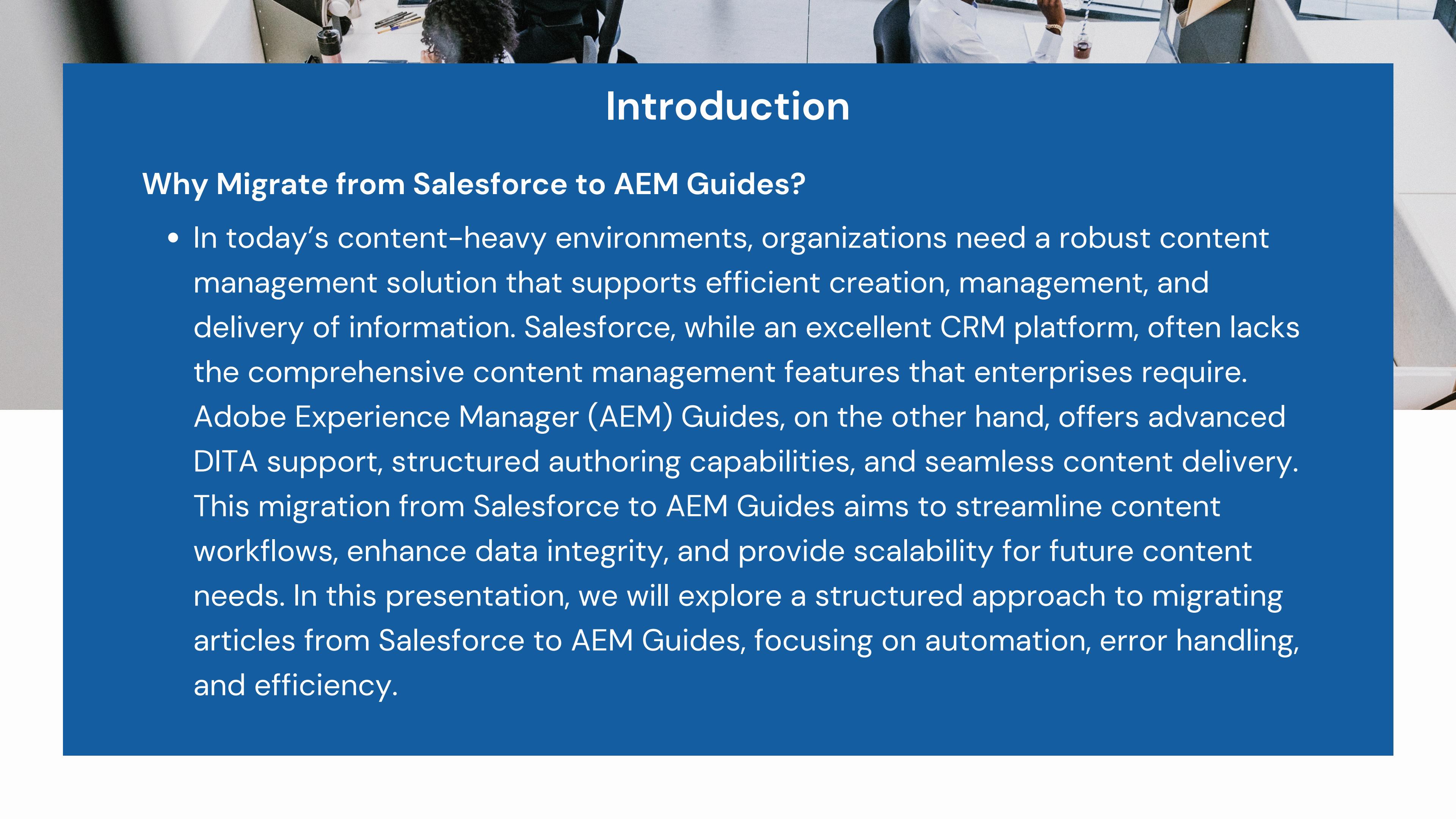
My expertise lies in full-stack development, blending both front-end and back-end technologies to deliver innovative and high-value solutions in every project.

With over 10 years of focused experience in development, I specialize in building robust and scalable applications, ensuring seamless integration across systems.

Table of content

- ▶ Introduction
- ▶ Step 1: Extracting Article Information
- ▶ Step 2: HTML Page Creation
- ▶ Step 3: Cleaning HTML and Updating Paths
- ▶ Step 4: HTML to DITA Conversion
- ▶ Step 5: Uploading Content into AEM Guides
- ▶ Step 6: Final Validation
- ▶ Advantages of this Migration Approach
- ▶ Conclusion





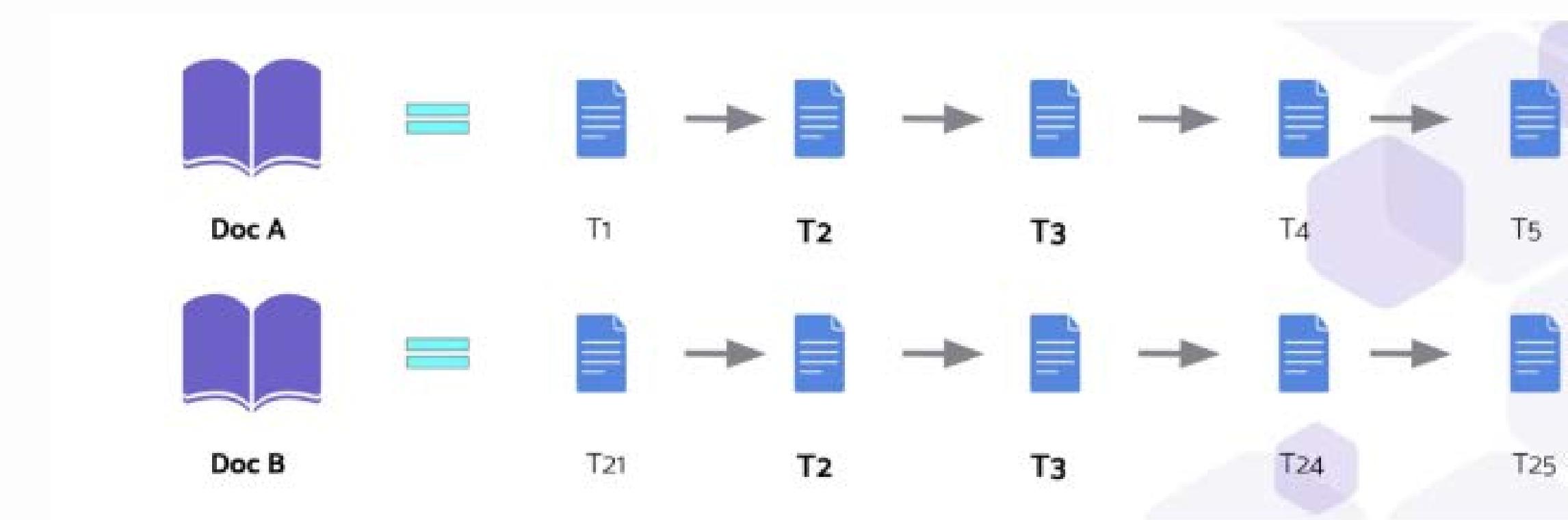
Introduction

Why Migrate from Salesforce to AEM Guides?

- In today's content-heavy environments, organizations need a robust content management solution that supports efficient creation, management, and delivery of information. Salesforce, while an excellent CRM platform, often lacks the comprehensive content management features that enterprises require. Adobe Experience Manager (AEM) Guides, on the other hand, offers advanced DITA support, structured authoring capabilities, and seamless content delivery. This migration from Salesforce to AEM Guides aims to streamline content workflows, enhance data integrity, and provide scalability for future content needs. In this presentation, we will explore a structured approach to migrating articles from Salesforce to AEM Guides, focusing on automation, error handling, and efficiency.

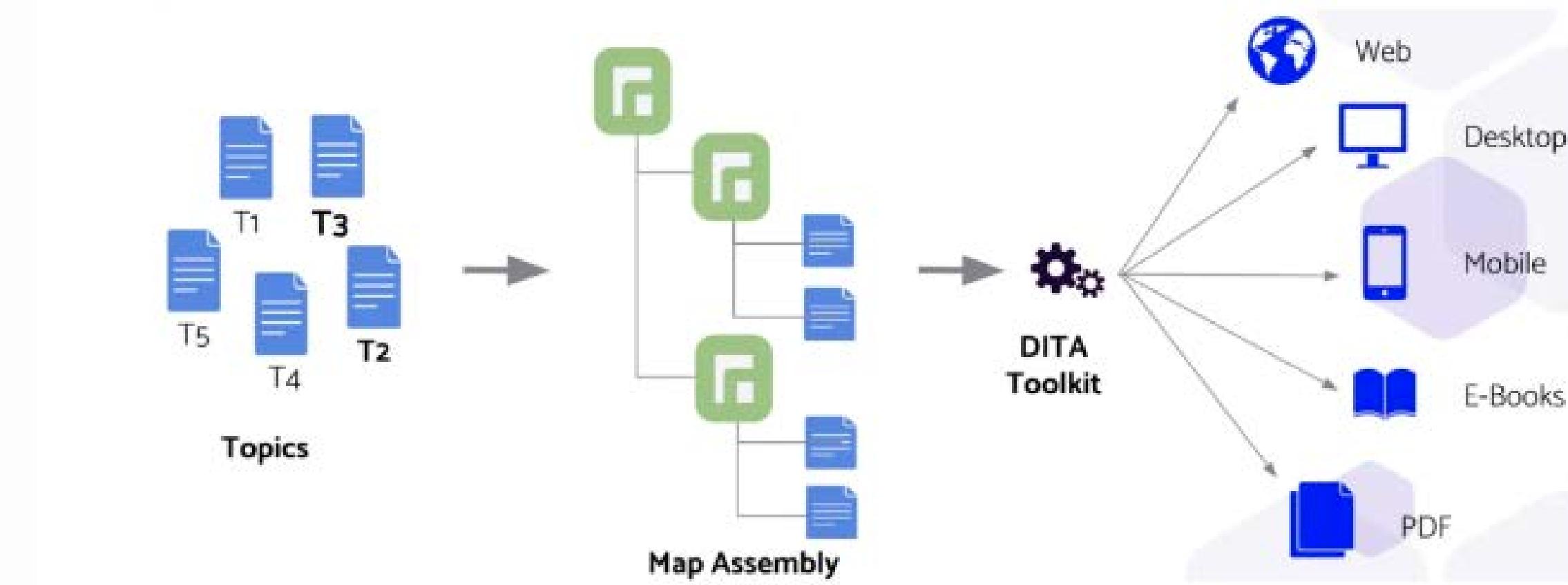
Linear Documentation without CCMS

- The linear approach to creating documentation has limitations where information is duplicated whenever we have to reuse it.
- Duplication reduces maintainability and adds overhead when the content changes.

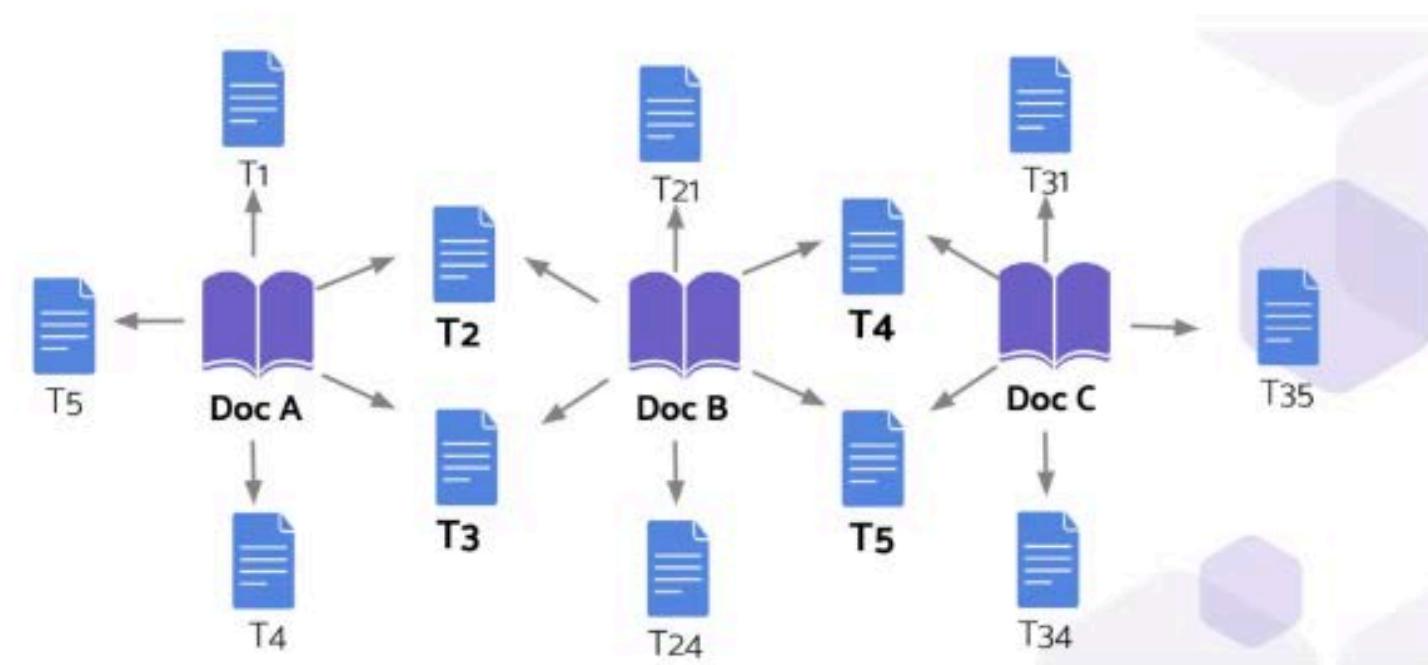
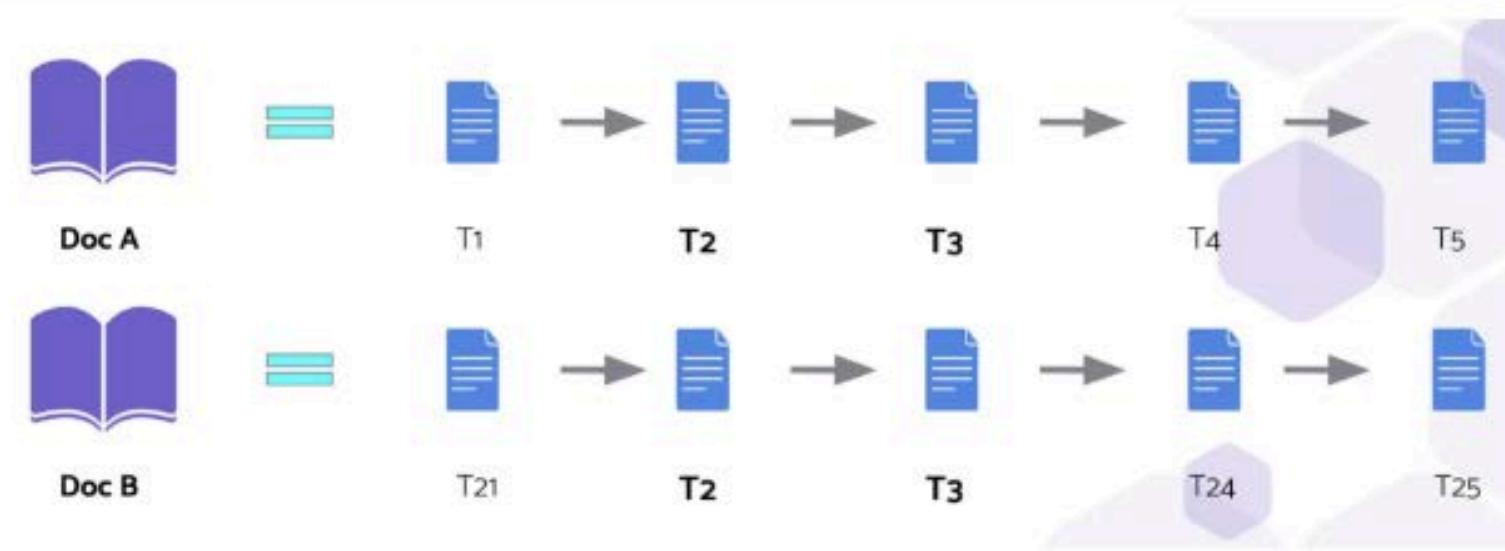


Component based documentation using CCMS

- CCMS allows reusing information by means of linking. The information is not duplicated but instead linked.
- Changes in the content reflect at all the places where the information is linked.



Linear Documentation vs CCMS



Step 1: Extracting Article Information

Data Extraction from Salesforce

- The first step in the migration process involves extracting article content from Salesforce.
- Using Selenium Wire, we automate the extraction of article titles, body content, and metadata fields that are essential for the migration process.
- Automation reduces the possibility of human error and speeds up the content extraction process.

Code Example: (Python)

```
from seleniumwire import webdriver  
  
# Initialize Selenium WebDriver  
driver = webdriver.Chrome()  
  
# Navigate to the Salesforce articles page  
driver.get('https://salesforce-articles-url.com')  
  
# Extract article information  
articles_info = []  
for article in driver.find_elements_by_class_name('article-class'):    title = article.find_element_by_class_name('title-class').text    content = article.find_element_by_class_name('content-class').text    articles_info.append({'title': title, 'content': content})  
  
driver.quit()
```

Step 2: HTML Page Creation

Creating HTML Pages for Articles

- After extracting data from Salesforce, the next step is creating individual HTML pages for each article.
- These HTML pages will serve as an intermediary format before converting the content into DITA.

Code Example: (Python)

```
import os

# Function to create HTML pages
def create_html_page(article):
    html_content = f"""
        <html>
            <head><title>{article['title']}</title></head>
            <body>{article['content']}
```

Step 3: Cleaning HTML and Updating Paths

Updating Paths and HTML Cleanup

- To ensure that all image and link paths work correctly in AEM, we need to update relative paths in the HTML code.
- The tool JTidy is then used to clean up the HTML and correct any markup errors, ensuring the HTML is well-formed and ready for conversion.

```
# Update paths for images and links
def update_paths(html_content):
    updated_content = html_content.replace('src="old-path/', 'src="new-path/')
    updated_content = updated_content.replace('href="old-path/', 'href="new-path/')
    return updated_content
```

Step 4: HTML to DITA Conversion

Converting HTML to DITA

- Now that the HTML is cleaned, the next step is to convert it into DITA format, which is essential for content in AEM Guides.
- This conversion allows us to take full advantage of AEM's structured content management features, enabling scalable and flexible content delivery.



```
def convert_html_to_dita(html_content):
    dita_content = f"""
        <topic>
            <title>{get_title_from_html(html_content)}</title>
            <body>{get_body_from_html(html_content)}</body>
        </topic>
    """
    return dita_content
```

Step 5: Uploading Content into AEM Guides



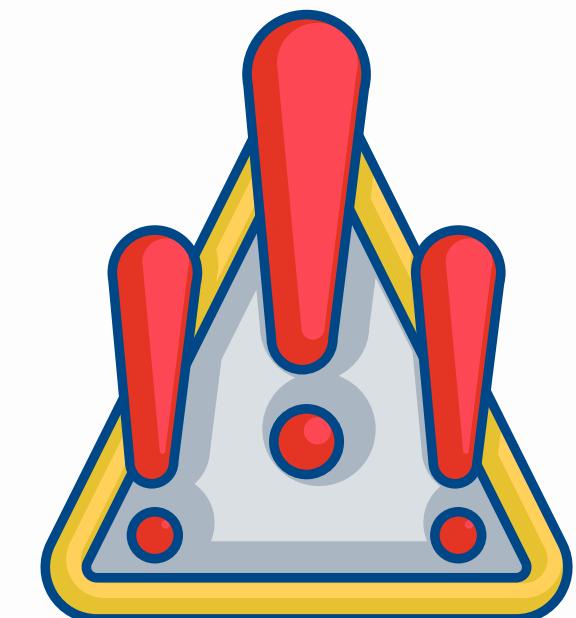
Seamless Upload to AEM Guides

- Once the articles have been converted into DITA, the next step is to upload the content into AEM Guides using acs commons Asset Importer.
- This process ensures that the content and metadata are correctly ingested into the AEM system for future use and content management.

Step 6: Final Validation

Final Validation and Error Handling

- The final step is validating that all articles, metadata, and assets have been correctly imported and are functioning as expected in the AEM Guides editor.
 - Error handling mechanisms ensure that if any issues arise during the migration, they are caught early, allowing for corrections before the migration is finalized.



Advantages of this Migration Approach

Key Benefits of Automated Migration

- Automation & Efficiency: By automating the process from extraction to final validation, we reduce the need for manual intervention and save significant time.
- Data Integrity: Each step of the migration is structured to maintain the integrity and consistency of the content.
- Scalability: The utility is designed to handle migrations of various sizes, from a few articles to thousands.
- Error Handling & Validation: Multiple validation points ensure that errors are detected early, minimizing risks.
- Cost Savings: By automating the entire migration process, the solution can potentially save substantial resources, resulting in significant cost reductions.

Conclusion

Migrating content from Salesforce to Adobe Experience Manager (AEM) Guides can bring numerous benefits to your organization. This migration utility reduces manual intervention, helps maintain data integrity, and offers a scalable solution that can adapt to the growing content needs of your enterprise. Furthermore, it supports better structuring and organization of your content through DITA, positioning your organization for long-term success in managing content within AEM Guides.

THANK YOU!



Bhanu Phanindra Babu

