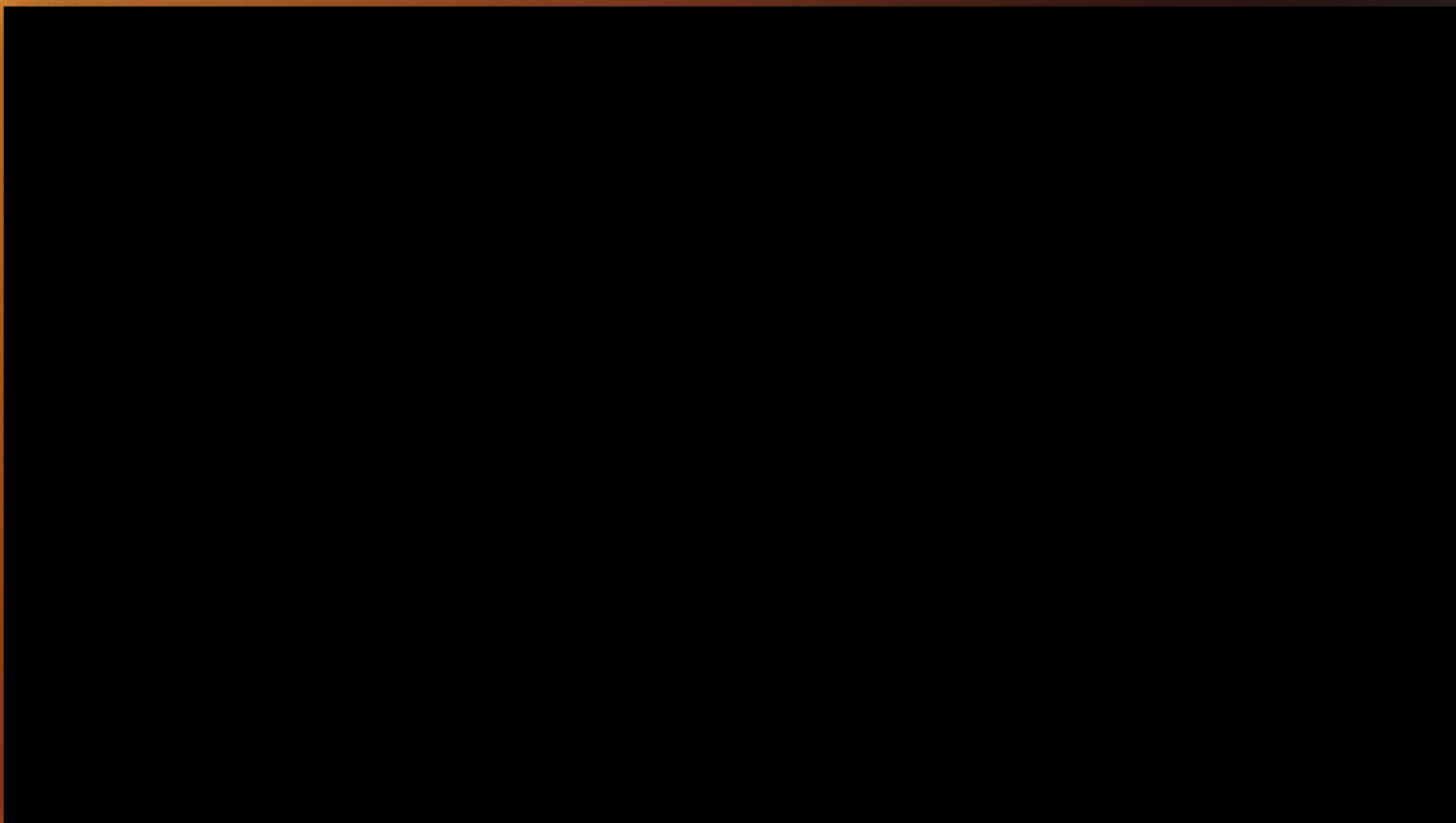


Developing a GenAI Extension for TYPO3

Insights from the AI Suite ✨





THE BIG PICTURE **AI SUITE**

www.autodudes.de

WHO'S BEHIND?

From TYPO3 lovers, for TYPO3 lovers.



ANDRÉ

andre@autodudes.de



MANU

manu@autodudes.de

THE AI SUITE AT A GLANCE.

The AI Suite combines everything for seamless AI workflows:

One API key for all
models and tools

Easy installation &
extension **setup**

Full TYPO3
integration

Automate tasks
and **Workflows**

Built-in tools
for **Accessibility**

SAAS, Whitelabel &
reseller options

Invoice-based
billing supported

GDPR compliant &
German infrastructure

Ready for **Public**
Sector Processes

ONE KEY. MANY POSSIBILITIES.

One central API key unlocks all (AI) capabilities:
text, image, translation, metadata, workflows...



OpenAI

ANTHROPIC



DeepL

AI Suite



OpenAI, Anthropic, Midjourney, Google Translate API, DeepL API
AI Suite (Meta Llama-3.3 (70B-Instruct))

MODELS & FEATURES

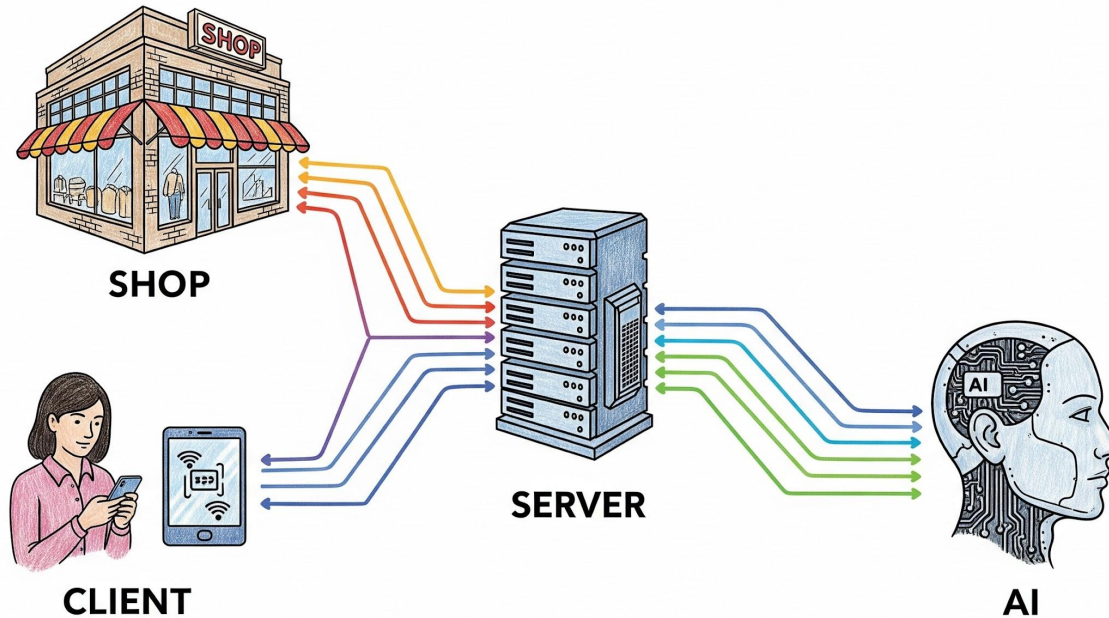
THREE DIFFERENT GENERATION MODES:

- text-to-text
- text-to-image
- image-to-text

ADDITIONALLY, NON AI-RELATED FEATURES:

- Permission-Management
- Prompt template manager
- Manually editing metadata in batch mode
- Asynchronous tasks

SYSTEM ORGANIZATIONAL CHART



ARCHITECTURE PRINCIPLES

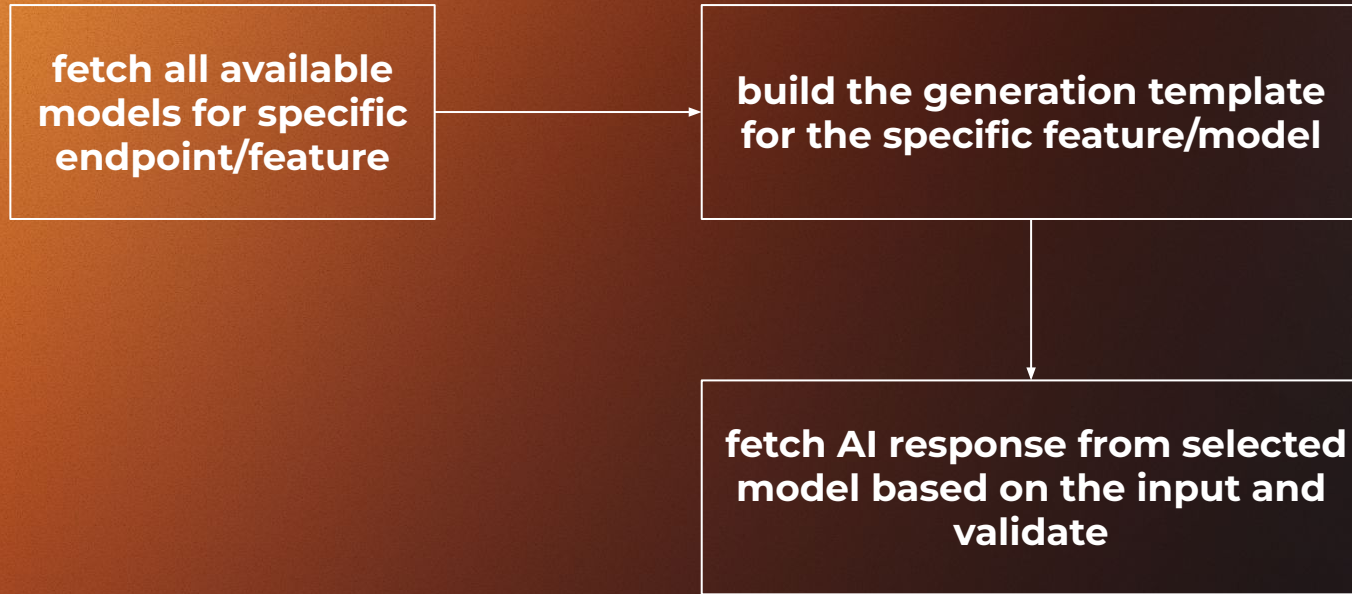
- MVC-Pattern
- Using PSR-14 Events and Hooks (where available/possible)
- Separating complex logic in services
- Using the DataHandler for creating/updating data
- Two main backend modules (for web and file area)
- Switched Extbase approach to Non-Extbase approach

MODEL INTEGRATION LAYER

- Flexible architecture for different models
- Activation / deactivation per backend user group
- Individual API keys per backend user group
- Scoping: e.g. for Metadata, Image, Content, Translate

MODEL INTEGRATION LAYER

GENERATION PROCESS



BACKEND CUSTOMIZATION - EVENTS

BeforePrepareConfigurationForEditorEvent:

CKEditor plugins

ModifyButtonBarEvent:

News generation, Image generation, AI translation (decorators: LocalizationController, DatabaseRecordList)

AfterPageTreeItemsPreparedEvent:

status indicators at bulk translation

ModifyPageLayoutContentEvent:

add "Whole page translation"

CustomFileControlsEvent:

add image generation buttons within FormEngine

RenderAdditionalContentToRecordListEvent:

"Whole page translation" action to list module

ModifyNewContentElementWizardItemsEventListener:

"AI Suite content" (NewContentElementController decorator)

BACKEND CUSTOMIZATION

HOOKS

Datahandler-Hook “processCmdmap_afterFinish”:
translation actions

\B13\Container\Hooks\Datahandler\CommandMapPostProcessingHook:
compatibility with EXT:container

MIDDLEWARE

Custom ParameterTrackingMiddleware:
remember filter settings

BACKEND CUSTOMIZATION

FRONTEND ARCHITECTURE

- **ECMAScript 6 modules + AJAX (~ 50 AJAX routes)**
- **MultiStepWizard approach**
- **Real-time UI feedback for credits/cost, status, notifications**



FACE THE HEAT
Challenges & Solutions

www.autodudes.de

LANGUAGE CHALLENGES

GOAL:


- Respect current language of editors work

CHALLENGE:

- Language information is not available at all places
- Considering language_uid -1 at identifying current language
- Considering in frontend disabled languages

LANGUAGE ID CHALLENGES

SOLUTION:

- Extending functionality with a central service “SiteService” 
- Adding additional selection options where language could not be automatically defined
 - e.g. Generation of metadata in filelist
 - e.g. Generation of new root page tree

LANGUAGE ID CHALLENGES

Dashboard

Pages

Agency

Prompt Manager

WorkflowManager

TaskEngine

WorkflowManager for file metas

Edit or update the meta properties of multiple files at once.

Selected directory: 1:/user_upload/ai-images/test/

Select field:


Image alternative text

☒ Show only empty fields

Show results


Found files with associated metadata

Select text generation AI

 Vision
1 Credit

☒

☐ Check / uncheck all

 finished african-elephant-in-the-wild.jpg

i Info

Select (target) language:

English [ai-suite-v13.ddev.site]

English [ai-suite-v13.ddev.site]

Chinesisch [ai-suite-v13.ddev.site]

Deutsch [autodudes-blog.ddev.site]

Deutsch [autodudes-blog.ddev.site]

Englisch [autodudes-blog.ddev.site]

Deutsch [ai-suite-v13.ddev.site]

Englisch [ai-suite-v13.ddev.site]

French [ai-suite-v13.ddev.site]

CKEDITOR PLUGIN CHALLENGES

GOAL:




- Modifying RTE content via AI
- Adding “Easy language” modification

CHALLENGE:

- Independent integration from local RTE configurations
- HTML-tags within RTE pose a challenge
- CKEditor is a third-party dependency
- Major change from CKEditor 4 to 5
- Considering language_uid -1 when identifying the current language

CKEDITOR PLUGIN CHALLENGES

SOLUTION:

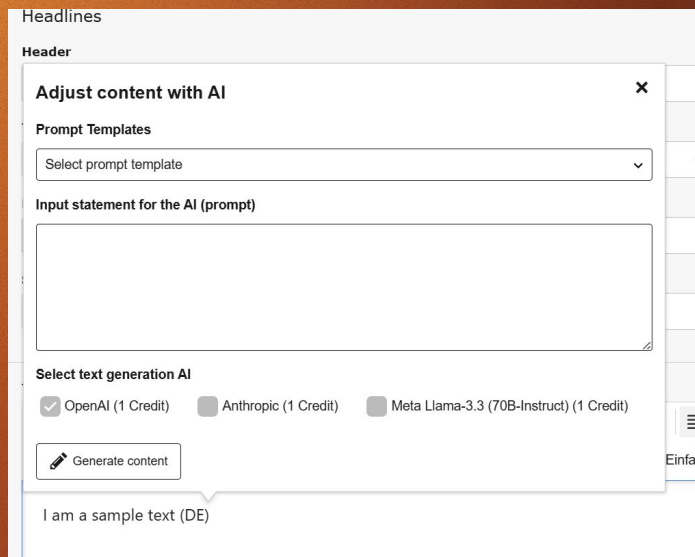
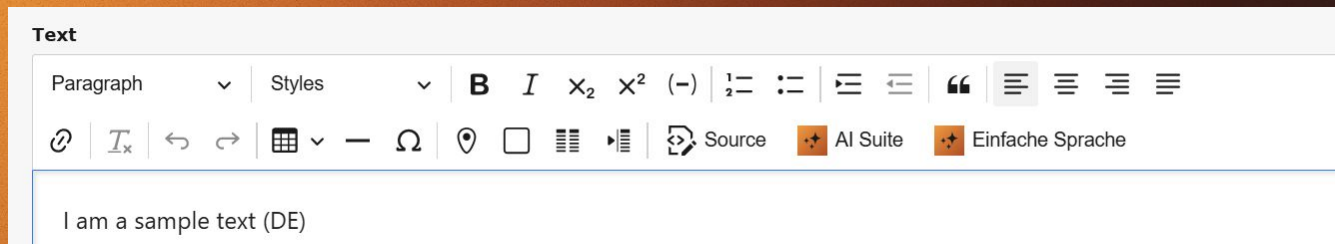
- Integrating through BeforePrepareConfigurationForEditorEvent 
- Implementing separate plugins for compatibility with different CKEditor versions
 - CKEditor version 4 (v11) 
 - CKEditor version 5 (since v12) 

Integration of CKEditor plugins via event

```
public function __invoke(BeforePrepareConfigurationForEditorEvent $event): void
{
    try {
        $langIsoCode = $this->siteService->getIsoCodeByLanguageId(
            (int)$event->getData()['databaseRow']['sys_language_uid'],
            $event->getData()['effectivePid']
        );
    } catch (Throwable $e) {
        return;
    }

    $this->pageRenderer->addInlineSetting( namespace: 'aiSuite', key: 'rteLanguageCode', $langIsoCode);
    $configuration = $event->getConfiguration();
    if ($this->backendUserService->checkPermissions( permissionsKey: 'tx_aisuite_features:enable_rte_aiplugin')) {
        $configuration['importModules'][] = [
            'module' => '@autodudes/ai-suite/ckeditor/AiPlugin/ai-plugin.js',
            'exports' => [
                'AiPlugin'
            ]
        ];
        $configuration['toolbar']['items'][] = 'AiPlugin';
    }
    // adding AiEasyLanguagePlugin to the RTE configuration
    ...
    $event->setConfiguration($configuration);
}
```

CKEDITOR PLUGIN CHALLENGES








TRANSLATION CHALLENGES

GOAL:

- Seamless integration with TYPO3 standard process
- Also including record translation

CHALLENGE FRONTEND PART:

- Different button integration in Page module  and List module 
- No events/hooks to use for
- Decorating classes is necessary  
- Decorating TYPO3 localization.js is necessary 
- Considering language_uid -1

TRANSLATION CHALLENGES

CHALLENGE BACKEND PART:

- Choosing an effective translation and fallback strategy
- Considering HTML- and FlexForm content
- Fetching relevant fields and content

TRANSLATION CHALLENGES

SOLUTION FRONTEND PART:

- Identifying and minimizing decorated classes and functions
- Adding necessary buttons to expected positions, leaving other elements/code unchanged

Adding additional translation information

```
public function localizeRecords(ServerRequestInterface $request): ResponseInterface
{
    $params = $request->getQueryParams();
    ...
    $this->process($params);
    return new JsonResponse([]);
}

@ Manuel Schnabel +1 *
protected function process($params): void
{
    $destLanguageId = (int)$params['destLanguageId'];
    $srcLanguageId = (int)$params['srcLanguageId'];
    $pageId = (int)$params['pageId'];
    ...
    $siteService = GeneralUtility::makeInstance( className: SiteService::class);
    $cmd['localization'][0]['aiSuite']['translateAi'] = str_replace( search: 'localize', replace: '', $params['action']);
    $cmd['localization'][0]['aiSuite']['srcLangIsoCode'] = $siteService->getIsoCodeByLanguageId($srcLanguageId, $pageId);
    $cmd['localization'][0]['aiSuite']['destLangIsoCode'] = $siteService->getIsoCodeByLanguageId($destLanguageId, $pageId);
    $cmd['localization'][0]['aiSuite']['destLangId'] = $destLanguageId;
    $cmd['localization'][0]['aiSuite']['srcLangId'] = $srcLanguageId;
    $cmd['localization'][0]['aiSuite']['uuid'] = $params['uuid'];
    $cmd['localization'][0]['aiSuite']['rootPageId'] = $siteService->getSiteRootPageId($pageId);
    ...
    $dataHandler = GeneralUtility::makeInstance( className: DataHandler::class);
    $dataHandler->start([], $cmd);
    $dataHandler->process_cmdmap();
}
```

Adding additional translation buttons to List module

```
public function makeLocalizationPanel($table, $row, array $translations): string
{
    $out = parent::makeLocalizationPanel($table, $row, $translations);
    if ($out && $this->backendUserService->checkPermissions(permissionsKey: 'tx_aisuite_features:enable_translation')) {
        $pageId = (int)($table === 'pages' ? $row['uid'] : $row['pid']);
        $possibleTranslations = $this->possibleTranslations;
        if ($table === 'pages') {
            $possibleTranslations = array_map(static fn ($siteLanguage) => $siteLanguage->getLanguageId(), $this->languagesAllowedForUser);
            $possibleTranslations = array_filter($possibleTranslations, static fn ($languageUid) => $languageUid > 0);
        }
        $languageInformation = $this->translateTools->getSystemLanguages($pageId);
        foreach ($possibleTranslations as $lUid_OnPage) {
            if ($this->isEditable($table)
                && !$this->isRecordDeletePlaceholder($row)
                && !isset($translations[$lUid_OnPage])
                && $this->getBackendUserAuthentication()->checkLanguageAccess($lUid_OnPage)
            ) {
                $out .= $this->translationService->buildTranslateButton(
                    $table,
                    $row['uid'],
                    $lUid_OnPage,
                    $this->listURL(),
                    $pageId,
                    $languageInformation[$lUid_OnPage]['flagIcon'],
                );
            }
        }
    }
    return $out;
}
```

Classes/Controller/RecordList/DatabaseRecordList.php

Adding same buttons through different events

```
public function __invoke(ModifyPageLayoutContentEvent $event): void
{
    $request = $event->getRequest();
    $pageLayoutContext = $this->createPageLayoutContext($request);

    if(!empty($pageLayoutContext->getNewLanguageOptions())) {
        $translationButtons = $this->backgroundTaskService->generateTranslationPageButtons($request);

        if (!empty($translationButtons)) {
            $event->addHeaderContent($translationButtons);
            $this->pageRenderer->loadJavaScriptModule( specifier: '@autodudes/ai-suite/translation/page-localization.js');
        }
    }
}
```

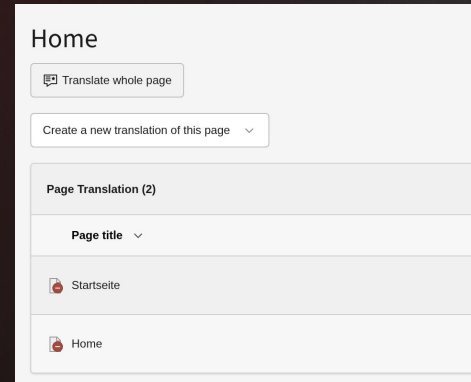
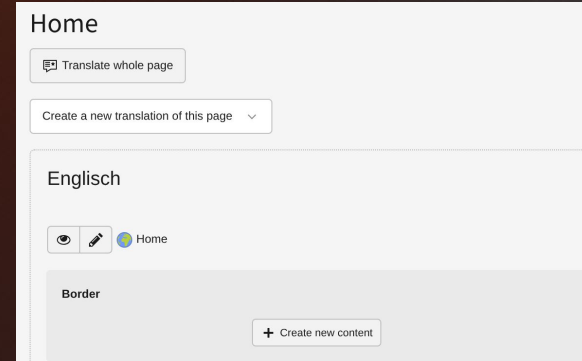
Classes/EventListener/ModifyPageLayoutContentEventListener.php

```
public function __invoke(RenderAdditionalContentToRecordListEvent $event): void
{
    $request = $event->getRequest();
    $pageLayoutContext = $this->createPageLayoutContext($request);

    if(!empty($pageLayoutContext->getNewLanguageOptions())) {
        $translationButtons = $this->backgroundTaskService->generateTranslationPageButtons($request);




        if (!empty($translationButtons)) {
            $event->addContentAbove($translationButtons);
            $this->pageRenderer->loadJavaScriptModule( specifier: '@autodudes/ai-suite/translation/page-localization.js');
        }
    }
}
```

Classes/EventListener/RenderAdditionalContentToRecordListEventListener.php



TRANSLATION CHALLENGES

SOLUTION BACKEND PART:

- Using DataHandler-Hook “processCmdmap_afterFinish” 
- Identifying FlexForm content and using the TYPO3 standard functionality in a separate Service 
- Using TYPO3 standard functionality of FormEngine 

Translating data in DataHandler hook

```
public function processCmdmap_afterFinish(DataHandler $dataHandler): void {  
    try {  
        if (isset($dataHandler->cmdmap['localization'][0]['aiSuite'])) {  
            $aiSuiteConfig = $dataHandler->cmdmap['localization'][0]['aiSuite'];  
  
            if ($this->isWholePageTranslation($aiSuiteConfig)) {  
                $this->processWholePageTranslation($dataHandler, $aiSuiteConfig);  
            } else {  
                $this->processSingleRecordTranslation($dataHandler, $aiSuiteConfig);  
            }  
        }  
    } catch (\Throwable $e) {  
        $this->logger->error('Error in TranslationHook: ' . $e->getMessage());  
        $this->addErrorFlashMessage();  
    }  
}
```

Fetching FlexForm data for translation purposes

```
public function convertFlexFormToTranslateFields(array $formData, array &$amp;translateFields): void
{
    $flexForm = $formData['databaseRow']['pi_flexform'] ?? '';
    if (empty($flexForm)) {
        return;
    }
    $newStructure = $formData["processedTca"]["columns"]["pi_flexform"]["config"]["ds"];
    $flexFormData = $flexForm['data'] ?? [];
    foreach ($flexFormData as $sKey => $sheetDef) {
        if (isset($newStructure['sheets'][$sKey]) && is_array($newStructure['sheets'][$sKey]) && is_array($sheetDef)) {
            foreach ($sheetDef as $lKey => $lData) {
                $this->checkValue_flex_procInData_travDS(
                    &dataValues: $flexFormData[$sKey][$lKey],
                    DSelements: $newStructure['sheets'][$sKey]['ROOT']['el'] ?? null,
                    structurePath: $sKey . '/' . $lKey . '/'
                );
            }
        }
    }
    $this->removeEmptyArraysRecursively(&$flexFormData);
    if(!empty($flexFormData)) {
        $translateFields['pi_flexform'] = [
            'data' => $flexFormData
        ];
    }
}
```

Fetching translatable fields and data

3 usages Manuel Schnabel +1

```
public function fetchTranslationFields(ServerRequestInterface $request, array $defaultValues, int $ceSrcLangUid, string $table): array
{
    $formData = $this->getFormData($request, $defaultValues, $ceSrcLangUid, $table);
    $translateFields = [];
    $this->getAllFieldsFromTableTypes($table, $formData, $translateFields);
    return $this->contentService->cleanupRequestField($translateFields, $table);
}
```

...

Manuel Schnabel


```
protected function getFormData(ServerRequestInterface $request, array $defaultValues, int $ceSrcLangUid, string $table)
{
    $formDataCompiler = GeneralUtility::makeInstance( className: FormDataCompiler::class);
    $formDataCompilerInput = [
        'request' => $request,
        'tableName' => $table,
        'vanillaUid' => $ceSrcLangUid,
        'command' => 'edit',
        'returnUrl' => '',
        'defaultValues' => $defaultValues,
    ];
    return $formDataCompiler->compile($formDataCompilerInput, GeneralUtility::makeInstance( className: TcaDatabaseRecord::class));
}
```


TRANSLATION CHALLENGES


Localize page "Translate" into French

Localization mode


Step 1 of 4




Content translations with OpenAI translate the content of the selected content elements into the selected target language.
One credit per 10,000 tokens (~ 7500 - 8000 words).




On average 2 times faster than OpenAI.
Content translations with Anthropic translate the content of the selected records/elements into the selected target language.
One credit per 5,000 tokens (~ 3500 - 4000 words).



On average 5 times faster than Anthropic, 10 times faster than OpenAI.
Content translations with GoogleTranslate translate the content of the selected records/elements into the selected target language.
One credit per 2,500 signs (~ 400 - 500 words).



On average 5 times faster than Anthropic, 10 times faster than OpenAI.
Content translations with DeepL translate the content of the selected records/elements into the selected target language.
One credit per 2,000 signs (~ 300 - 350 words).



Content translations with Meta Llama-3.3 (70B-Instruct) translate the content of the selected content elements into the selected target language.
One credit per 5,000 tokens (~ 3500 - 4000 words)

Cancel Previous Next

Record & Whole Page Translation




























Über uns

Translate whole page

Create a new translation of this page

Page (3) >

New Page Download Show columns

Page title	Localization	Localize to	Description
<input type="checkbox"/> Geschichte	    	Deutsch    	
<input type="checkbox"/> Team	    	Deutsch    	
<input type="checkbox"/> Mitgliedschaft	    	Deutsch    	

BATCH PROCESSING CHALLENGES

GOAL:







- Processing asynchronous tasks with huge workload
- Saving maximum editor work time

CHALLENGE:

- Asynchronous processing without Scheduler
- PHP limitations on different infrastructures
- HTTP Basic Authentication with special characters
- Handling backend user permissions
- Files & file references: filtering empty, filtering only referenced files

BATCH PROCESSING CHALLENGES

SOLUTION:

- Separate Backend Modules: Web & File 
- Integrating a step-by-step sending process (maximum 5 pages/files) per request 
- Considering checks for PHP `post_max_size` & `max_upload_size` 
- Considering HTTP Basic Authentication, Base64 encoding 
- Using the TYPO3 standard approach from the Filelist for checking file permissions 
- Using the `sys_ref_index` table to identify included files 

BATCH PROCESSING CHALLENGES

Dashboard

Pages

Agency

Prompt Manager

WorkflowManager

TaskEngine

TaskEngine

Overall Statistics

Total	Finished	Pending	Error
31	31	0	0

Detailed Statistics

Task List

Filter Tasks

File metadata TaskEngine

Options

☐ Click and Save

Refresh

Delete all visible tasks

Image title (title) 31




finished  african-elephant-in-the-wild.jpg

Image title  




finished  cats-planning-mice-catching-seo-strategy.jpg

Image title  

UX CHALLENGES



GOAL:

- Remembering selected filter options

CHALLENGES:

- Switching the context leads to filter reset
- Filter information get lost

SOLUTION:

- Tracking filter settings via Middleware 
- Choosing settings stored in the user session 

CONTENT ELEMENT GENERATION CHALLENGES

GOAL:




- Generating different available content elements with AI

CHALLENGES:

- Identifying relevant content elements and fields
- Handle conflicting events to consider additional configuration (like PageTS)
- Modifying results in RTE (RTE configuration out-of-the-box only available in FormEngine)
 - CKEditor 4 (v11)
 - CKEditor 5 (since v12)

CONTENT ELEMENT GENERATION CHALLENGES

SOLUTION:

- Using TCA to identify selectable content elements and fields 
- Using a similar approach by decorating the `NewContentElementController` (identical integration to TYPO3 standard, just using another route) 
- CKEditor: Building an own `RichtextElementService` that adapt the relevant parts of needed configuration 
 - there are open tasks left

CONTENT ELEMENT GENERATION CHALLENGES

New Page Content

Search for any content type

Menu15

Form elements2

Special elements3

Container6

News system9

Data2


Interactive5

Media11


Social Media1

Text13


AI Suite Content24

 **Accordion (AI Suite)**


Insert several Accordionelement(s) with headline and text (with AI generated content)

 **Carousel Fullscreen (AI Suite)**


Insert Carouselement with different contents (header / text with image / pure HTML) (with AI generated content)

 **Timeline (AI Suite)**


Insert a element to display a timeline (with AI generated content)

 **Gallery (AI Suite)**


Display multiple images in a rich gallery. (with AI generated content)

 **Images Only (AI Suite)**


Any number of images aligned in columns and rows with a caption. (with AI generated content)

 **Tab (AI Suite)**


Insert several Tabelement(s) with headline and text (with AI generated content)

 **Text & Images (AI Suite)**


Any number of Images wrapped right

 **Carousel (AI Suite)**


Insert Carouselement with different contents (header / text with image / pure HTML) (with AI generated content)

 **Carousel Small (AI Suite)**


Insert Carouselement with different contents (header / text with image / pure HTML) (with AI generated content)

 **Card Group (AI Suite)**


Insert a element with several content cards (with AI generated content)

 **Icon Group (AI Suite)**


Insert a element with several icons (with AI generated content)

 **Media (AI Suite)**

Any number of youtube and vimeo videos in columns and rows with a caption. (with AI generated content)

 **Text & Media (AI Suite)**

Any number of media wrapped right around a regular text element. (with AI generated content)

 **File Links (AI Suite)**

Makes a list of files for download. (with AI

CONTENT ELEMENT GENERATION CHALLENGES

Create new content of type "text"

Select text generation AI



OpenAI
1 Credit



Anthropic
1 Credit

Select image generation AI

A maximum of **two images** will be created in the course of the first request. Additional images can then be added at any time.



DALL-E
3 Credits



Midjourney
4 Credits

General fields

Available input fields

☒ Header ☒ Subheader

☒ Text

Available file fields

☒ Background Image

Prompt Templates

Select prompt template

Input statement for the AI (prompt)

Give me some example intro text for the TYPO3 Developer Days 2025



Content is being generated, this may take a moment...

Cancel

Started image generation, image 1

AI (prompt)

Intro text for the TYPO3 Developer Days 2025

CONTENT ELEMENT GENERATION CHALLENGES

[illegible]

Fetching RTE configuration

```
public function fetchRteConfig(array $data): array
{
    ...
    $ckeditorConfiguration = $this->resolveCkEditorConfiguration($data);
    $ckeditorAttributes = GeneralUtility::implodeAttributes([
        'id' => $fieldId . 'ckeditor5',
        'options' => GeneralUtility::jsonEncodeForHtmlAttribute($ckeditorConfiguration, useHtmlEntities: false),
    ], xhtmlSafe: true);
    $textareaAttributes = GeneralUtility::implodeAttributes([
        'slot' => 'textarea',
        'id' => $fieldId,
        'name' => $itemFormElementName,
        'rows' => '18',
        'class' => 'form-control',
        'data-formengine-validation-rules' => $this->getValidationDataAsJsonString($config),
    ], xhtmlSafe: true);
    ...
    return [
        'ckeditorAttributes' => $ckeditorAttributes,
        'textareaAttributes' => $textareaAttributes
    ];
}
```

COST & TRANSPARENCY CHALLENGES

GOAL:


- Transparency about costs and composition of credits usage per action

CHALLENGES:


- Different actions have different costs and compositions of credits
- Credits depends on tokens (translations of contents)
- Quick overview about used and left credits
- Different pricing of available models


COST & TRANSPARENCY CHALLENGES

SOLUTION:

- Add the costs directly near the action buttons and update them based on user selection
- For translation process, we add the composition and ranges into the explanation
- Custom ToolbarItem to show the current state 

COST & TRANSPARENCY CHALLENGES

 **DALL-E**
3 Credits

 **Midjourney**
4 Credits

General fields

Available input fields

☒ Header☒ Subheader

☒ Text


Available file fields

☒ Background Image

Prompt Templates

Select prompt template

Input statement for the AI (prompt)


 Generate content

(4 Credits)


Localize page "Translate" into French

Localization mode


Step 1 of 4

 OpenAI


Content translations with OpenAI translate the content of the selected content elements into the selected target language.
One credit per 10,000 tokens (~ 7500 - 8000 words).

 Anthropic


On average 2 times faster than OpenAI.
Content translations with Anthropic translate the content of the selected records/elements into the selected target language.
One credit per 5,000 tokens (~ 3500 - 4000 words).

 Google Translate

On average 5 times faster than Anthropic, 10 times faster than OpenAI.
Content translations with GoogleTranslate translate the content of the selected records/elements into the selected target language.
One credit per 2,500 signs (~ 400 - 500 words).

 DeepL

On average 5 times faster than Anthropic, 10 times faster than OpenAI.
Content translations with DeepL translate the content of the selected records/elements into the selected target language.
One credit per 2,000 signs (~ 300 - 350 words).

 Meta Llama-3.3 (70B-Instruct)

Content translations with Meta Llama-3.3 (70B-Instruct) translate the content of the selected content elements into the selected target language.
One credit per 5,000 tokens (~ 3500 - 4000 words).

Cancel

Previous

Next

TESTING & QS CHALLENGES

GOAL:

- Test coverage for critical and important actions/processes

CHALLENGES:

- Testing interaction between systems
- Testing TYPO3 functionality and Server/APIs functionality
- Testing different major versions of TYPO3
- Testing TYPO3 backend with E2E tests
- Testing AI-APIs (non-deterministic response)
- Handling iFrame structure and dynamically changing selectors in the TYPO3 backend

TESTING & QS CHALLENGES

SOLUTION:

- Separation of TYPO3 + server and API functionality tests
- Mocking requests and response data (deterministic responses)
- Standardizing tests and separating version-specific parts
- Implementing separate AI-API tests
- Implementing helper functions, e.g. for fetching UUIDs in FormEngine
- Currently ~ 500 tests

→ Playwright demo repository 



WHAT WE EXPECT FROM **TYPO3 GenAI Toolbox**

www.autodudes.de

TAKEAWAYS

WHAT WE HAVE LEARNED OUR WAY:

- AI is moving fast, TYPO3 Core has a release cycle of 1.5 years
- AI interfaces are not standardized across all providers
- TYPO3 core can not support every individual LLM API. This means, we need a standardized registration process

THOUGHTS TO FACE THE CHALLENGES

POTENTIAL SOLUTIONS:

- TYPO3 can provide interfaces or abstract classes
- Provide a registration, e.g. via Dependency Injection
- Automatic registration, e.g. via scope (text, translation, image, metadata-page, metadata-file, CKEditor)
- MCP server: get content, get TCA config, get page structure
- Normalize user interaction in MultiStepWizard, (backend module) or CKEditor plugin
- Global instructions and prompt templates
- Glossary for translation models
- Permission handling

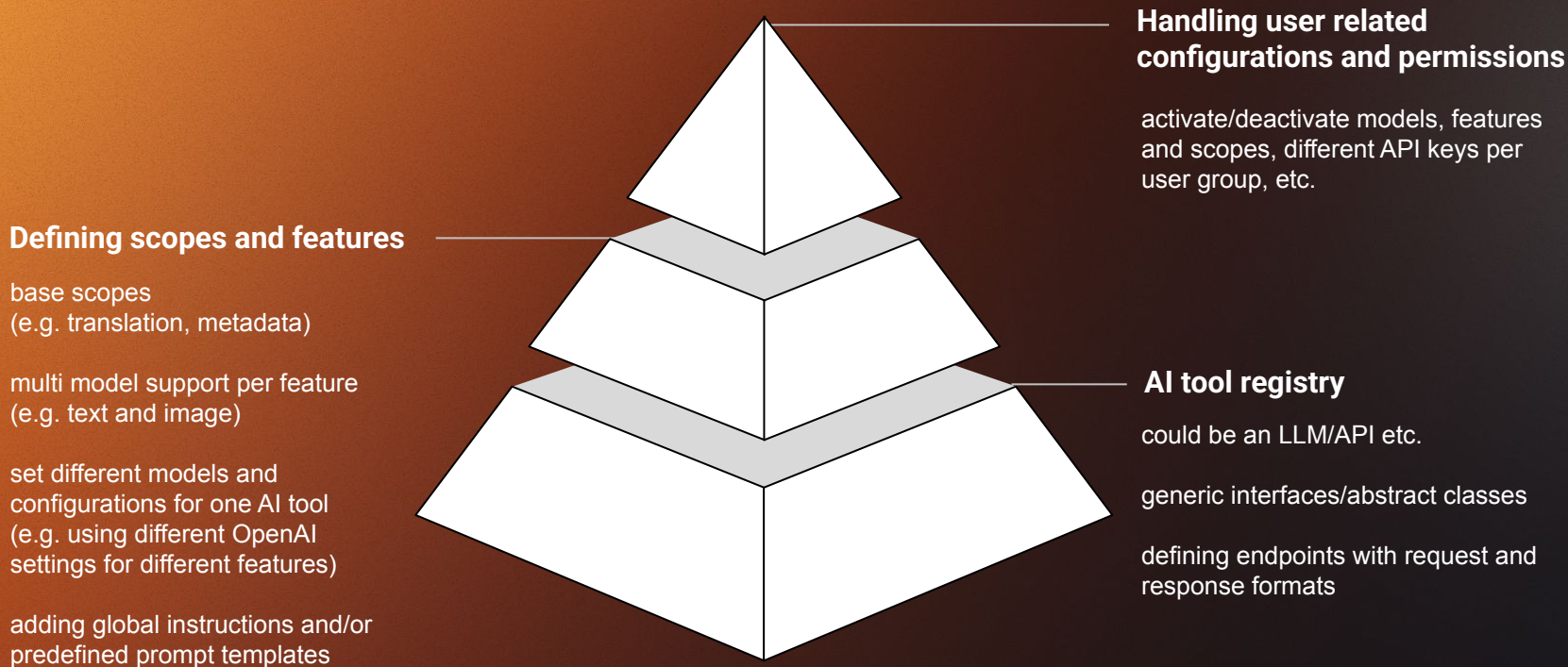
THOUGHTS TO FACE THE CHALLENGES

POTENTIAL SOLUTION, INTERFACE OR ABSTRACT CLASS

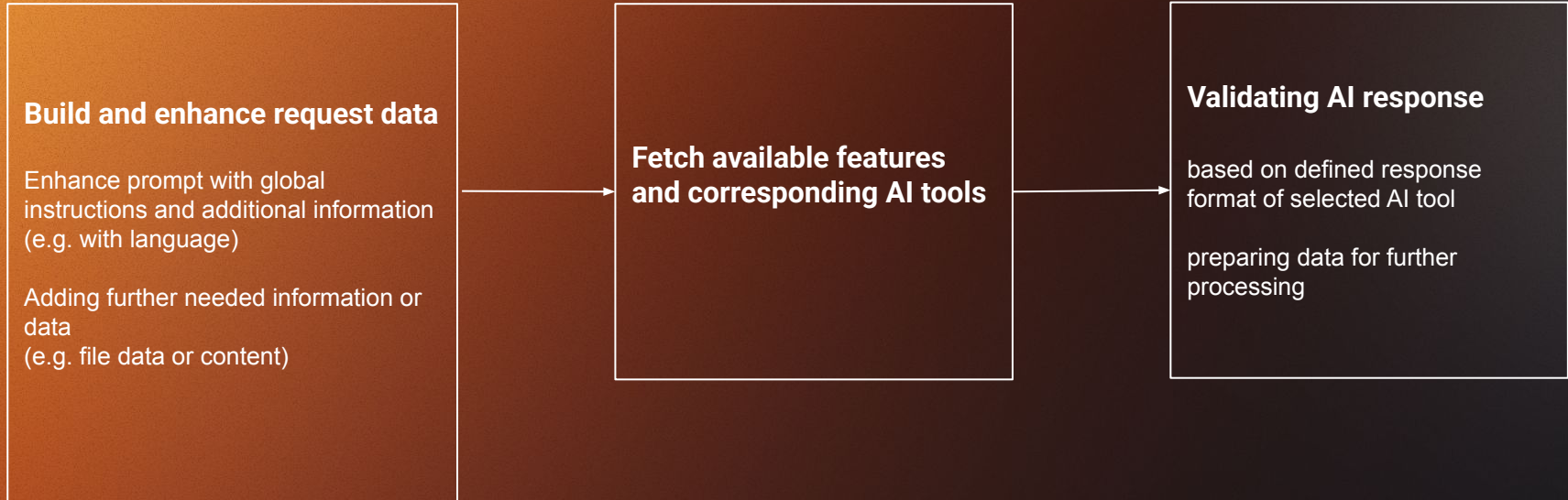
- Scope (where to register, e.g. in backend?)
- Configuration (registration of the API settings)
- Process method (sends user input to API)
- Workflow processing (e.g. MultiStepWizard steps with defined input fields)
- Title, description, icon, costs(?)

MODEL ABSTRACTION LAYER

POTENTIAL SOLUTION



HANDLING AI GENERATION PROCESS



LET'S BUILD THE FUTURE OF TYPO3 TOGETHER!



FOLLOW US
ON SOCIAL MEDIA
@autodudes_ai



JOIN OUR
SLACK CHANNEL.
#aisuite



THANKS,
DUDES!

www.autodudes.de