



ODC 20
26

Omnis Developer Conference

AI in Omnis Apps

How to use AI in Omnis applications
– practical examples, use cases
and integration approaches

Maurin Wannags,
Omnis Software Germany GmbH



Contents

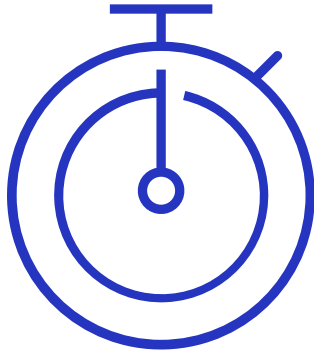
AI in Omnis Apps – From Idea to Implementation

- Benefits of an AI Integration
- Basic Principles of an AI Integration
- Possible Integration Approaches in Omnis Studio
- Hands-On Example: AiERP
- Detailed Implementation in the AiERP Project
- Tips & Lessons Learned

Benefits of an AI integration

AI in Omnis Apps – From Idea to Implementation

Benefits of an AI Integration



- Improved user experience.
- Chatbots and AI assistants help with day-to-day tasks and speed up work.
- No need for complex search forms or filters.
 - *“Show me all outstanding invoices over €5,000 from the last quarter.”*

AI in Omnis Apps – From Idea to Implementation

Benefits of an AI Integration



- Interpretation of complex data
 - *Development of forecasts (pattern recognition; e.g., which customers are at risk of churning?)*
 - *Assists in decision-making*
- Quick integration of new KPIs

AI in Omnis Apps – From Idea to Implementation

Benefits of an AI Integration

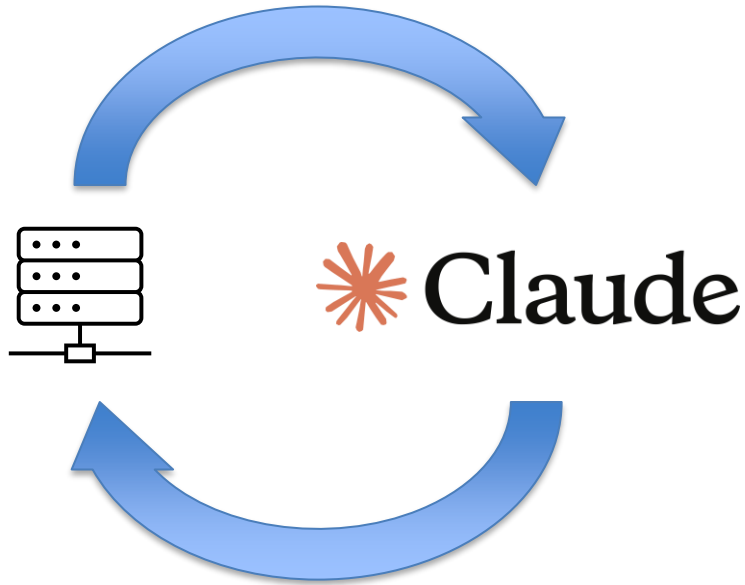


- Users are used to AI chats/features in their daily lives (ChatGPT, Claude, Copilot, etc.)
- Industry solutions with AI integration stand out.
- This serves as a relevant selling point.
- It demonstrates innovation.

Basic Principles of an AI Integration

AI in Omnis Apps – From Idea to Implementation

Basic Principles of an AI Integration



- The Omnis app sends a request (HTTP request) to an AI service.
- The AI service sends a response back.

AI in Omnis Apps – From Idea to Implementation

Basic Principles of an AI Integration – Request/Response-Cycle

- In the case of more complex tasks, several cycles of requests and responses are usually required:
 - *The application sends a request to the AI service*
 - *The AI service responds*
 - *The application processes the response and, if necessary, sends an automated response containing data or results back to the AI*
 - *The AI responds again (the cycle restarts as needed)*
 - *In some cases, the AI may call multiple methods to retrieve the necessary arguments*

AI in Omnis Apps – From Idea to Implementation

Basic Principles of an AI Integration – Request

Header

System
Prompt

Body

- The header contains authentication information, usually in the form of an API key
- Most AI providers support a system prompt that includes basic information about the context of the request
 - *Employee/username*
 - *Date and time*
 - *Brief description of the system in which the AI is integrated*
- Standard “request body” with content or prompt

AI in Omnis Apps – From Idea to Implementation

Basic Principles of an AI Integration – Tool Use

```
37 Line: {
38 Line:   "name": "create_appointment",
39 Line:   "description": "Creates a new entry or appointment in the calendar.",
40 Line:   "input_schema": {
41 Line:     "type": "object",
42 Line:     "properties": {
43 Line:       "start": {
44 Line:         "type": "string",
45 Line:         "description": "Start of the appointment: Date and time with time zone in i
MM:SS+HH:MM, e.g., 2026-05-07T15:00:00+02:00"
46 Line:       },
47 Line:       "end": {
48 Line:         "type": "string",
49 Line:         "description": "End of appointment: Date and time with time zone in the fc
SS+HH:MM, e.g., 2026-05-07T15:00:00+02:00"
50 Line:       },
51 Line:       "title": {
52 Line:         "type": "string",
53 Line:         "description": "Title of the appointment"
54 Line:       },
55 Line:       "description": {
56 Line:         "type": "string",
57 Line:         "description": "Brief description of the appointment. Approx. 20 words"
58 Line:       }
59 Line:     },
60 Line:     "required": ["date", "start", "end", "title"]
61 Line:   }
62 Line: }
63 Line: }
```

- Supported by most AI providers
 - *Anthropic: "Tool use"*
 - *Gemini: "Function calling"*
- Developers define a schema of functions
 - *Name, description, input schema (parameters), required fields*

AI in Omnis Apps – From Idea to Implementation

Basic Principles of an AI Integration – Tool Use

```
# call the correct function
Switch pFunctionName
  Case "check_remaining_vacation_days"
    Do method checkremainingvacationdays (pParams.employee) Returns IJSON
  Case "get_employee"
    Do method getemployee (pParams) Returns IJSON
  Case "create_appointment"
    Do method createappointment (pParams) Returns IJSON
  Default
  End Switch

Quit method IJSON
```

- The AI response contains a “tool_use” content block.
- The custom-created functions are then called via a user-defined “parsing” of the response.
- The application returns the result of the called functions to the AI.

Possible Integration Approaches in Omnis Studio

AI in Omnis Apps – From Idea to Implementation

Possible Integration Approaches in Omnis Studio

- Chatbot without internal functionality (general-purpose Q&A only)
- Categorization of data within the existing workflow
- AI Assistant Chatbot with “tool_use”
 - *Predefined methods*
 - *Dynamic SQL methods*
- ...

AI in Omnis Apps – From Idea to Implementation

Possible Integration Approaches in Omnis Studio – Predefined Methods

- A method that returns a result (true/false or data)
- Basically like any other method
- Limited to the defined scope
 - *AI can combine several methods on a case-by-case basis*

AI in Omnis Apps – From Idea to Implementation

Possible Integration Approaches in Omnis Studio – Dynamic SQL Methods

- A precondition is that the AI knows the database structure (system prompt)
- The AI receives the initial prompt and dynamically generates an SQL statement
- The application receives the SQL statement, executes it, and sends the result back to the AI
- The AI formulates the response and returns it to the application
- **Caution: The application should check the SQL statement before executing it!**
 - *E.g.: read-only*
 - *Prohibited keywords*
 - *Locked tables/columns*
 - *LIMIT*

Hands-On

Example: AiERP

AI in Omnis Apps – From Idea to Implementation

Hands-On Example: AiERP

- AI integration in the form of an assistant
- Limitations due to predefined methods
- Powered by Anthropic's "claude-sonnet-4-6"
- Use of the OW3HTTPWorker

AI in Omnis Apps – From Idea to Implementation

Hands-On Example: AiERP

Starts at 3pm and goes for two hours

Got it! Here are the details I have so far: - **Title:** Meeting - **Date:** 2026-05-16 - **Start:** 3:00 PM - **End:** 5:00 PM Could you also provide me with: 1. **Title** – What is the meeting about? 2. **Time zone** – Is it also UTC+0?

Yes please


Could you also let me know what the meeting is about so I can give it a proper title? 😊

Title can be just Meeting

Here are the full details: - **Title:** Meeting - **Date:** 2026-05-16 - **Start:** 3:00 PM (UTC+0) - **End:** 5:00 PM (UTC+0) Shall I go ahead and create this appointment?

Please

Your meeting has been successfully added to your calendar! 📅 - **Date:** May 16, 2026 - **Time:** 3:00 PM – 5:00 PM (UTC+0) Is there anything else I can help you with? 😊




Demonstration

Detailed Implementation in the AiERP Project

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

- The chat saves the user's request (as a message and as JSON in a separate context table)

	id [PK] integer 	json text
1	166	{ "role": "user", "content": "Hi I have an dental appointment tomorrow at 1 pm"}
2	167	{ "role": "assistant", "content": [{ "type": "text", "text": "I'd be happy to create a call" }
3	168	{ "role": "user", "content": "It takes about an hour and +0"}

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

Do method `definertools` Returns `IToolsJSON`
Do method `definesystemprompt` Returns `ISystemPromptJSON`
Do method `buildmessagecontext` (`pContext`) Returns `IContextJSON`

- Assembling the HTTP request

Begin text block

```
Line:{  
Line: "model": "claude-sonnet-4-6",  
Line: "max_tokens": 4096,  
Line: "system": [ISystemPromptJSON],  
Line: "tools": [IToolsJSON] ,  
Line: "messages": [IContextJSON]  
Line:}
```

End text block

Get text block `IBodyJSON`
Quit method `IBodyJSON`

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

Begin text block

Line: [

For `pContext.$line` from 1 to `pContext.$linecount`

Line: [`pContext.json`][`pick(bool(pContext.$line = pContext.$linecount), ",", "")`]

End For

Line:]

End text block

Get text block `IContextJSON`

Quit method `IContextJSON`

buildmessagecontext

Begin text block

Line: "You are an ERP assistant at this company. The current user is [`con($ctask.tUserObject.iUser.lastname, ", ", $ctask.tUserObject.iUser.firstname)`], user id: [`$ctask.tUserObject.iUser.id`]. The current timestamp is [`jst(#D, "D: y-M-D")`]. Answer questions politely and concisely in English. When performing actions that involve data entry (creating an appointment, modifying an invoice, etc.), always ask for confirmation before executing the action."

End text block

Get text block `ISystemPrompt`

Quit method `ISystemPrompt`

definesystemprompt

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

- Initializing and starting the OW3 HTTP Worker

Do method **fetchcontext** Returns **IContext**

Set reference **iChatObject** to **pChatObjectReference**

Do method **buildrequest** (**IContext**) Returns **IBodyJSON**

Do method **defineheader** Returns **IHeader**

Do **\$cinst.\$init**(**cAnthropicURL**,**kOW3httpMethodPost**,**IHeader**,**IBodyJSON**) Returns **ok**

Do **\$cinst.\$start** () Returns **ok**

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

- Save the AI response in the context table

```
Do IContent.role.$assign(pResponseContent.role)
```

```
Do IContent.content.$assign(pResponseContent.content)
```

```
Do OJSON.$formatjson(OJSON.$listorrowsToJson(IContent)) Returns IJSON
```

```
Do iChatObject.$savecontext(IJSON) Returns IHistory
```

```
Quit method IHistory
```

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

- Iterate through AI response blocks and perform the corresponding actions

```
Do IResponseContent.content.$cols.$first() Returns IColRef
While IColRef
  Do IResponseContentContent.$assign(IResponseContent.content.[IColRef.$ident])
  Switch IResponseContentContent.type
    Case "text"
      Do iChatObject.$sendmessage(IResponseContentContent.text,kTrue,IHistory.id)
    Case "tool_use"
      Do method tooluse (IResponseContentContent.name,IResponseContentContent.input) Returns IJSON
      Do method buildtoolresult (IResponseContentContent.id,IJSON) Returns IToolResult
      # add result to list of tool use results
      Do IToolResults.$add(IToolResult)
    Default
  End Switch
  Do IResponseContent.content.$cols.$next(IColRef) Returns IColRef
End While
```

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

The image displays four overlapping screenshots of a 'Variable IResponse...' dialog box, illustrating different stages or types of AI responses in the AiERP project.

Top Left Screenshot: Shows a full message object with the following data:

model	claude-sonnet-4-6
id	msg_01DxNcRaUfiY3yZ1H7hzB3DZ
type	message
role	assistant
content	(Not empty)
stop_reason	tool_use
stop_reason	NULL
stop_details	NULL
usage	1050

Top Middle Screenshot: Shows a text response:

type	text
text	Let me check that for you right away!

Bottom Middle Screenshot: Shows a list of text items:

__1	text
__2	(Not empty)

Bottom Right Screenshot: Shows a tool use object with the following data:

type	tool_use
id	toolu_01Mnot17ZNb15aSFVXyrWaN9
name	check_remaining_vacation_days
input	(Not empty)
caller	(Not empty)

AI in Omnis Apps – From Idea to Implementation

Detailed Implementation in the AiERP Project

- Save the application's responses as JSON content in the context table and restart the AI request process
- The AI will then process the “tool_results” and respond accordingly

```
If IResponseContent.stop_reason="tool_use"  
  # add tool use results silently (without visible message) to histroy  
  Do method savetoolresults (IToolResults) Returns IToolHistory  
  # process history  
  Do iChatObject.$callai()  
End If
```

Tips & Lessons Learned

AI in Omnis Apps – From Idea to Implementation

Tips & Lessons Learned

- A good system prompt makes a huge difference
 - *Specify the language*
 - *Describe the context clearly*
 - *If necessary, represent the database structure, etc.*
- In Omnis Studio, do as much work as possible with lists and rows, then reformat them into JSON (OJSON: \$listorrows tojson(), \$formatjson())
- Where data protection is important:
 - *Use GDPR-compliant providers*
 - *Host AI models yourself (if necessary, via services like RunPod or Google Vertex AI)*



Thank you very much!