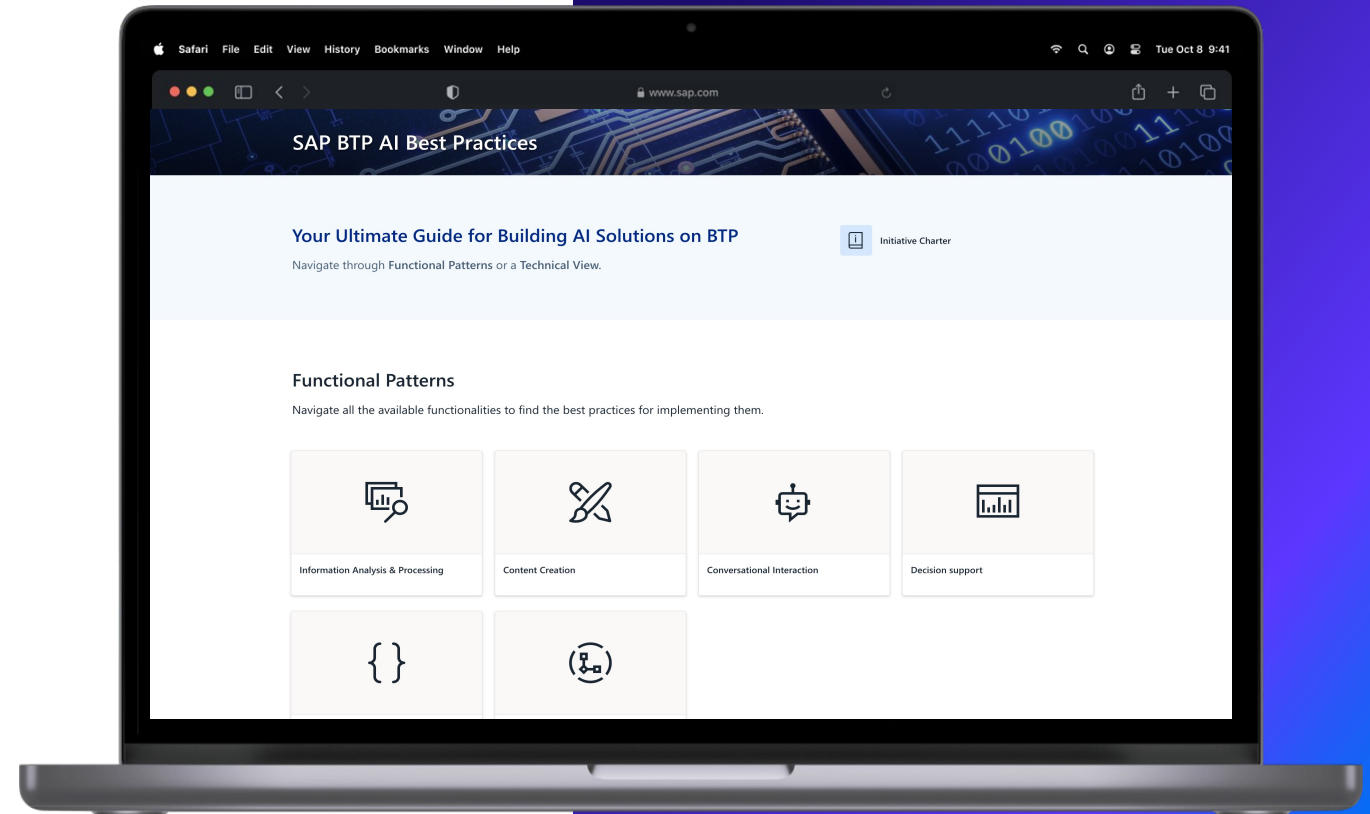


SAP BTP AI Best Practices

Access to Generative AI Models

A structured approach to efficiently **integrate AI models into business applications** using SAP AI Core.



BTP AI Services Center of Excellence

24.03.25

Steps

- 1 Overview**
- 2 Pre-requisites**
- 3 Key Choices and Guidelines**
- 4 Implementation**

Access to Generative AI Models

Simple Consumption of Generative AI Models

Interacting with deployed Generative AI models involves **sending requests** to **retrieve generated responses**.

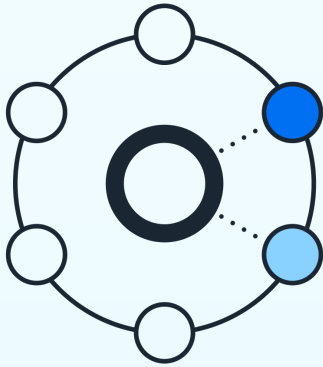
In the request, we can define the **prompt**, the **model**, and the **parameters** that control the response generation.

Expected Outcome

- Provide secure and efficient access to Generative AI models.
- Enables applications to leverage the power of AI models and provide a wide spectrum of functionalities.

Key Benefits

Why use BTP AI Core instead of direct access?



Model Interchangeability

Easily switch between models with the same code and implementation to best suit your task.



Out-of-the-box Features

Supercharge your development built-in capabilities like Data Masking, Prompt Templating, Filtering, and more.

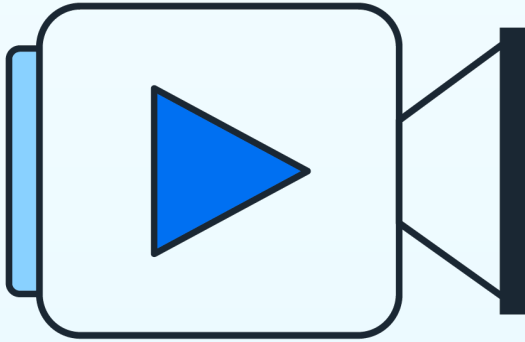


Security & SAP Ecosystem

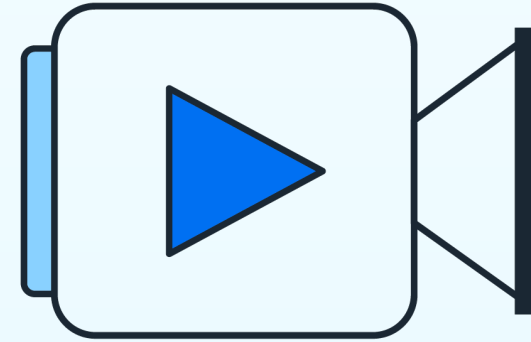
It's fully integrated into the SAP Ecosystem, leveraging the best of SAP technologies.

Video Content

Access to Generative AI Models



1-minute Teaser



In-depth Webinar

Pre-requisites

Business

- SAP AI Core with the “Extended” tier on SAP BTP ([Pricing Information](#))

Technical

- SAP Business Technology Platform (SAP BTP) subaccount ([Setup Guide](#))
 - SAP AI Core ([Setup Guide](#))
-

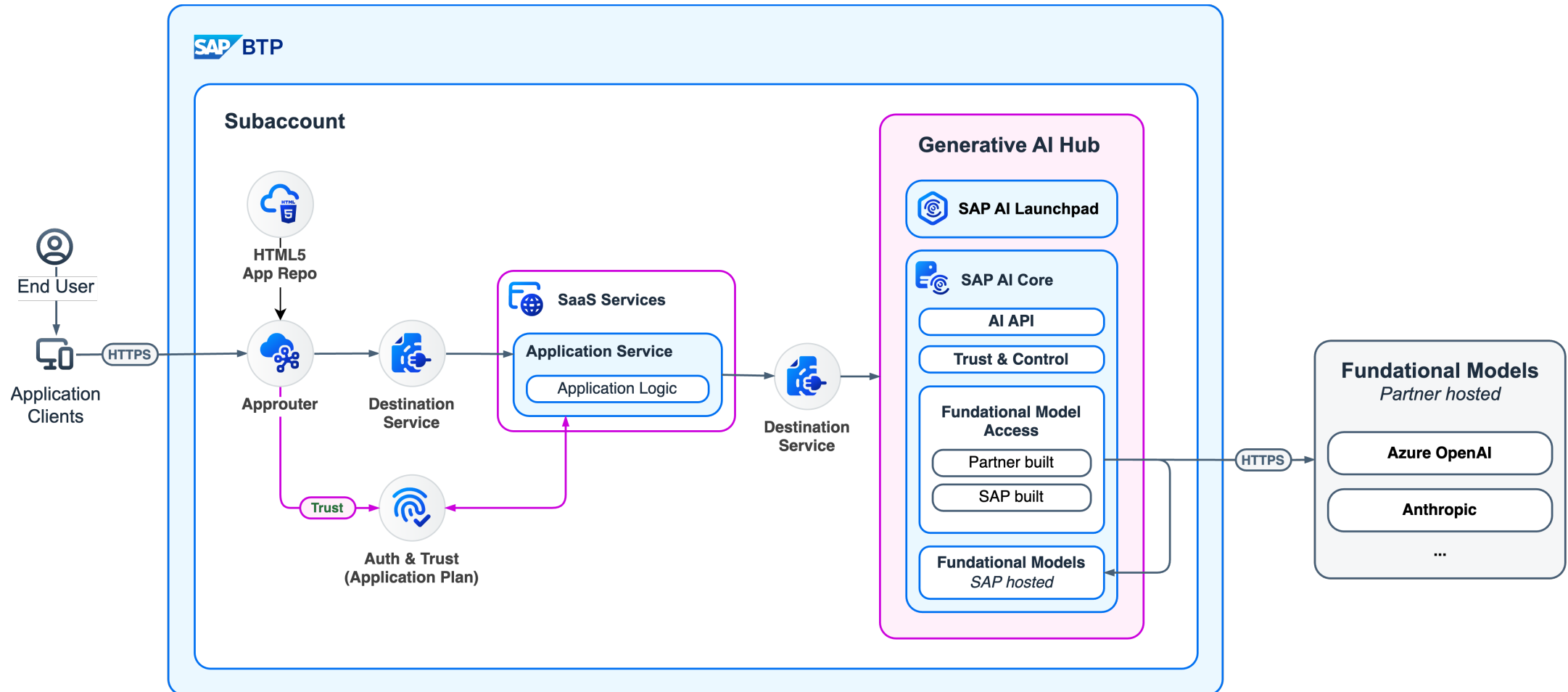
SAP Business Technology Platform (SAP BTP)

- SAP Business Technology Platform (BTP) is an integrated suite of cloud services, databases, AI, and development tools that enable businesses to build, extend, and integrate SAP and non-SAP applications efficiently.

SAP AI Core

- SAP AI Core is a managed AI runtime that enables scalable execution of AI models and pipelines, integrating seamlessly with SAP applications and data on SAP BTP that supports full lifecycle management of AI scenarios.

High-level reference architecture



Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Model

- The choice of model is critical to the **accuracy** of the answer, **operating costs**, **language**, and **context window**.
- Larger models are usually better for more complex tasks involving reasoning, longer context window, and better inference.

How to choose the model?

- Use the [Model Library](#) to inform your model choice using benchmarking data.

Relevant Links

- List of available models
- AI Core Endpoint

Generative AI / Model Library

Model Library

Explore available models and their specifications. Inform your model choice using benchmarking data.

Filters

×

↺

Capabilities:

☐ Embedding

☐ Image Generation

☐ Image Recognition

☐ Speech To Text

☒ Text Generation

Input Types: ↺

Access Type: ↺

Model Provider: ↺

Included Types: ↺

☒ **Deprecated**

Models (32)

Mode:

Catalog

Leaderboard

Chart

Search Name

Rank foundation models using benchmarks.

Model	Helm Lite Mean Win Rate	ChatBotArena Arena Score	AirBench Refusal Rate	Output Token Cost (Capacity Units)
ABAP Codestral Version: v1 (latest)				
ABAP Starcoder2-7b Version: v4 (latest)				
Claude 3 Haiku Version: 1 (latest)	0.263	1179	0.827	1.99394
Claude 3 Opus Version: 1 (latest)	0.683	1247	0.844	110.06987
Claude 3 Sonnet Version: 1 (latest)	0.377	1201	0.847	22.13495
Claude 3.5 Sonnet Version: 1 (latest)	0.885	1268	0.859	22.13495
Falcon-40b Instruct Version: null (latest)				
GPT-3.5 Turbo Version: 0613	0.358	1117	0.631	2.73326
GPT-3.5 Turbo Version: 1106 (latest)		1068	0.525	2.73326
GPT-3.5 Turbo 16k Version: 0613 (latest)				5.33210
GPT-4 Version: 0613	0.867	1163	0.642	77.56196
AI-1.3.2 Instruct Version: 0613 (latest)	0.983	1183	0.643	11.28788
AI-1.3.2 Instruct Version: 0613 (latest)				2.33370
AI-1.3.2 Instruct Version: 0613 (latest)		1068	0.632	5.13358
AI-1.3.2 Instruct Version: 0613 (latest)	0.328	1113	0.631	5.13358
AI-1.3.2 Instruct Version: 0613 (latest)				

Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Settings

- Picking correct values for temperature, max tokens, frequency penalty, and presence penalty is crucial to the quality of the inference.

Model Parameters

Frequency Penalty (Reduces repetition of frequent words)

- High – **Summarization**: Reduces redundancy in reports or articles.
- Low – **Poetry/Mantras**: Encourages repetition for stylistic effect.

Presence Penalty (Discourages reuse of any previous words)

- High – **Idea Brainstorming**: Promotes diverse, non-repetitive topics.
- Low – **Themed Storytelling**: Reinforces a central topic or character.

Max-Tokens (Limits response length)

- Low – **Chatbot Responses**: Short, to-the-point answers.
- High – **Long-Form Content**: Detailed articles or stories.

Temperature (Controls randomness of output)

- Low – **Technical/Legal Writing**: Precise, deterministic output.
- High – **Creative Writing**: More imaginative and random text.

Model Configuration

Selected Model

gpt-4o



Selected model may contain content filters that apply in addition to user settings.

Parameters (5)



Frequency Penalty:



Presence Penalty:



Max Tokens:



Temperature:



Streaming Response:



Streaming Response:

Implementation

Programming Model Selection Guidelines

Backend-Only API

Use **Python** (well-maintained) or **JavaScript/TypeScript** (strong async capabilities, Node.js ecosystem).

Full-stack Application (UI & Backend)

Use **CAP App** for optimized performance, scalability, and seamless SAP integration.

Python

SDK

- [SAP Generative AI hub SDK](#) (For building apps)
- [SAP AI Core SDK](#) and [AI API Client SDK](#) (AI Core lifecycle)

Reference Code

- [SAP BTP AI Best Practices - Sample Code](#)
- [BTP Gen AI Hub SDK Samples \(sample #2\)](#)

Learning Journeys

- [Consumption of GenAI models Using Orchestration - A Beginner's Guide](#)

JavaScript/TypeScript

SDK

- [SAP Cloud SDK for AI](#)

Reference Code

- [SAP BTP AI Best Practices - Sample Code](#)
- [SAP Cloud SDK for AI - Sample Code \(orchestration file\)](#)

Learning Journeys

- [Consumption of GenAI models Using Orchestration - A Beginner's Guide](#)

CAP App

SDK

- [SAP Cloud SDK for AI](#) (Recommended)
- [CAP LLM Plugin](#)

Reference Code

- [SAP BTP AI Best Practices - Sample Code](#)
- [SAP Cloud SDK for AI - Sample Code \(orchestration file\)](#)

Learning Journeys

- [Consumption of GenAI models Using Orchestration - A Beginner's Guide](#)

Java

SDK

- [SAP Cloud SDK for AI \(for Java\)](#)

Reference Code

- [SAP BTP AI Best Practices - Sample Code](#)
- [Sample Spring App example \(Service file and Controller file\)](#)

Learning Journeys

- [Consumption of GenAI models Using Orchestration - A Beginner's Guide](#)

Code Sample

JavaScript/TypeScript

```
const orchestrationClient = new OrchestrationClient({
  // Define the language model to be used
  llm: {
    model_name: 'gpt-4o',
    model_params: {
      max_tokens: 1000, // Maximum number of tokens to generate in the response. This limits the length of the generated text.
      temperature: 0.6, // Lower values make the output more deterministic, while higher values make it more random.
      n: 1 // Number of responses to generate.
    }
  },
  // Define the prompt
  templating: {
    template: [{ role: 'user', content: 'What is the capital of France?' }]
  }
});

// Execute the request
const result = await orchestrationClient.chatCompletion();

console.log(result.getContent());
```

Contributors



Marques, Luis



CIGAINA, MARCO



Rzhakysynskyi, Andrii



Stoyanov, Velizar



Rodriguez, Joel



Chopra, Aman



Bartler, Felix



Robledo, Francisco



Schulz, Robert



Antonio, Dan



Humphries, Ian



Lange, Joerg

Thank you