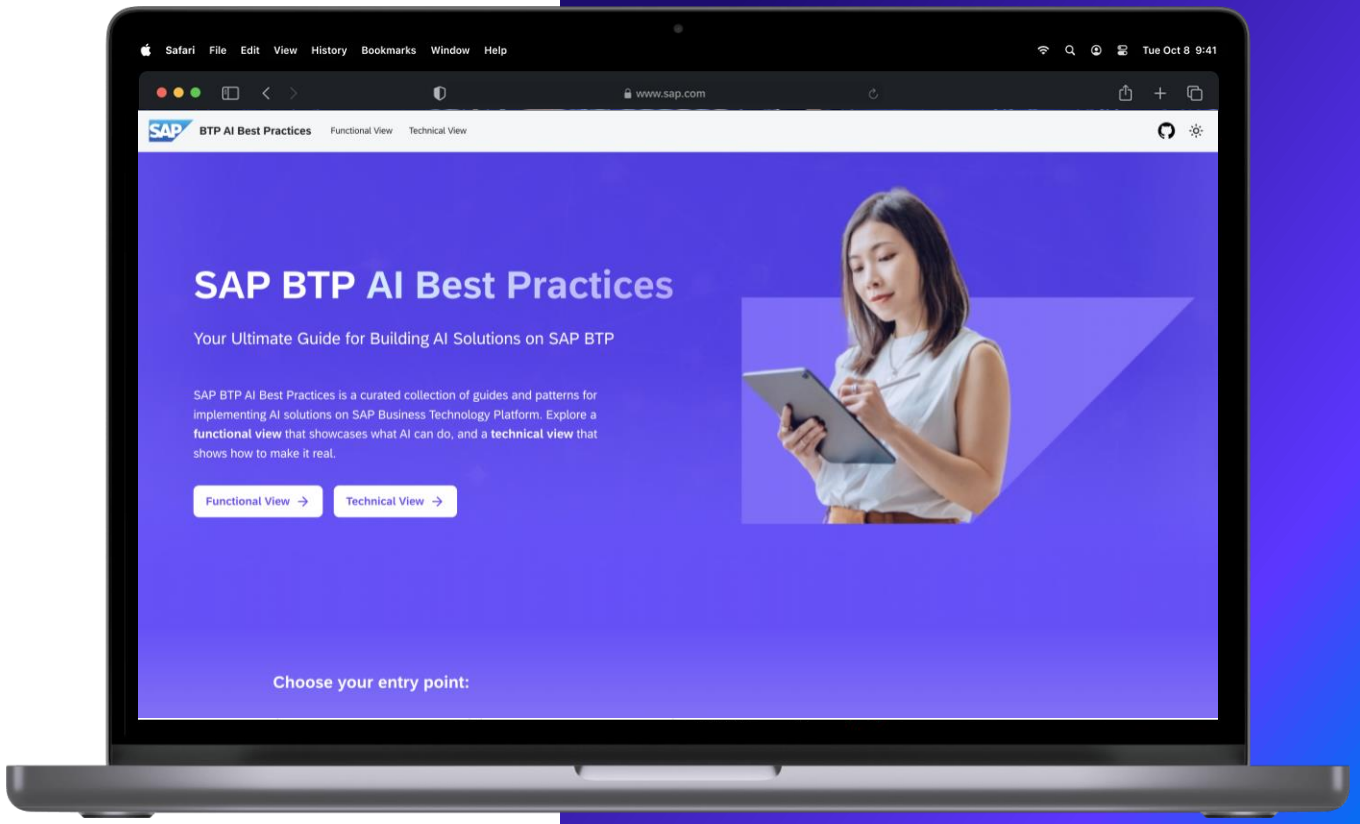
SAP BTP AI Best Practices

Content filtering

A structured approach to efficiently **control and manage the content** that is processed by and generated from generative AI Models.



BTP AI Services Center of Excellence

12.05.2025

Steps

1 Overview

2 Pre-requisites

3 Key Choices and Guidelines

4 Implementation

Content filtering

Define content filters to prevent undesirable behavior

A critical process that ensures the **quality** and relevance of **input and output data.**

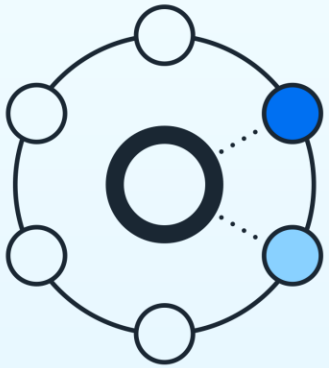
This process is **essential** for maintaining the **accuracy** and **reliability** of LLMs, especially when they are used for **real-time applications** or **sensitive tasks.**

Expected Outcome

- AI models in SAP AI Core produce safe, appropriate and relevant content.
- AI models adhere to the specified content safety criteria.

Key Benefits

Why use BTP AI Core instead of direct access?



Prevents Harmful or Inappropriate Outputs

Content filtering helps AI systems avoid generating or spreading offensive, violent, hateful, or sexually explicit content, which is especially critical in public-facing applications like chatbots or content generation tools.

Compliance and Brand Safety

For businesses using AI in customer service, marketing, or user-generated content platforms, filtering helps prevent reputational damage and legal issues by ensuring that AI-generated content aligns with company policies and legal regulations.

Ethical and Responsible AI Use

By enforcing guardrails around what AI models can produce or respond to, content filtering helps align AI behavior with ethical guidelines, community standards, and cultural sensitivities.

Pre-requisites

Business

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

Technical

1. Set up an SAP Business Technology Platform (SAP BTP) subaccount (Setup Guide)
2. Deploy SAP AI Core with extended service plan (Setup Guide)
3. Configure the Orchestration service in AI Launchpad (Setup Guide)

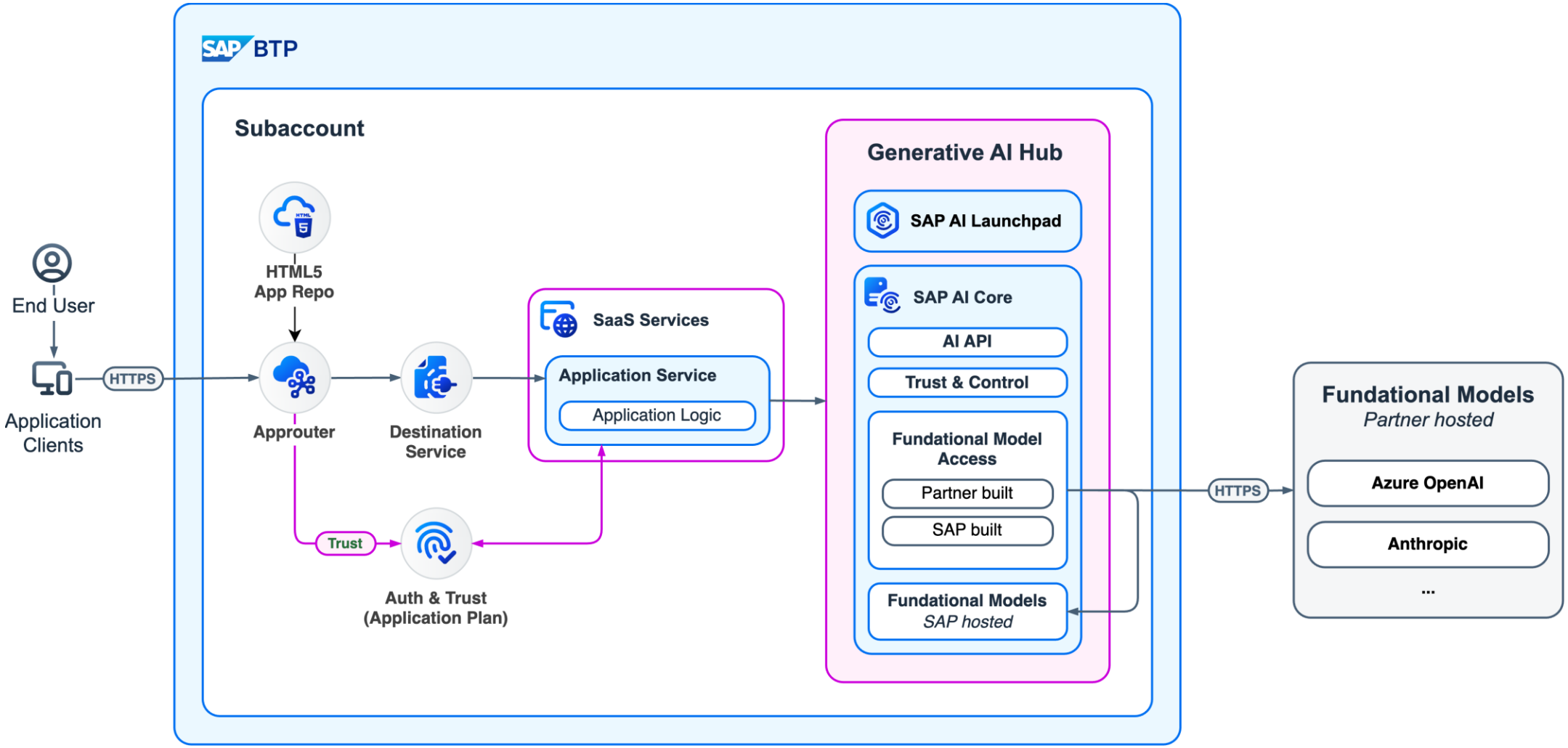
SAP Business Technology Platform (SAP BTP)

- 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

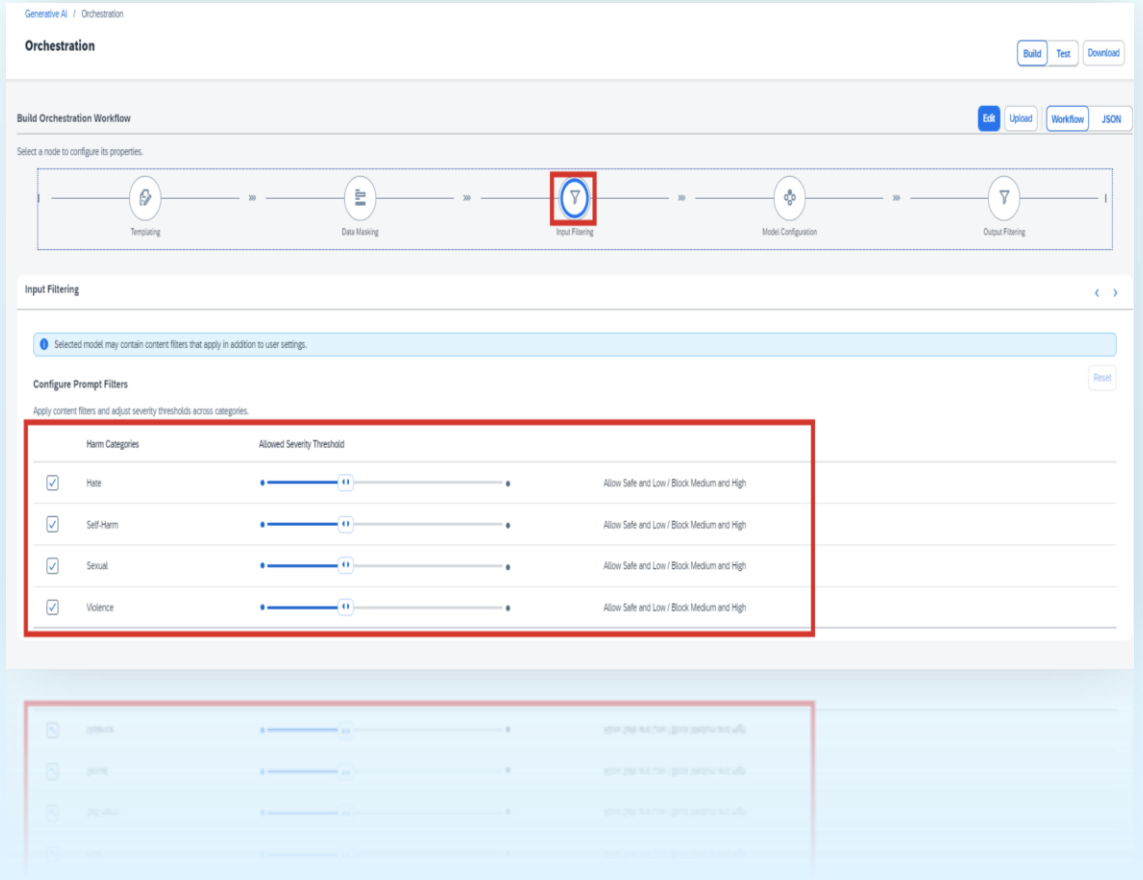


Key Choices and Guidelines

Decisions that impact the performance and utility of the application

Filtering guidelines

- **Filtering selection:** Selecting the appropriate filtering method and criteria in accordance with your use case and organization values is key to responsible AI.
- **Ethical considerations:** Strive to eliminate bias, unfair and discriminatory content.
- **Ensure Real-Time Performance:** Implement filtering that is fast and efficient, avoiding noticeable delays in AI responses.
- **Maintain Ongoing Monitoring and Updates:** Regularly review and update filtering policies and datasets to keep up with evolving language and threats.
- **Transparency and Explainability:** Provide clear reasons for filtering decisions and include user feedback mechanisms to build trust and support compliance.



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

- Leveraging Orchestration Capabilities to Enhance Responses

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

- Leveraging Orchestration Capabilities to Enhance Responses

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

- Leveraging Orchestration Capabilities to Enhance Responses

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

- Leveraging Orchestration Capabilities to Enhance Responses

Code Sample

JavaScript/TypeScript

```typescript
export async function orchestrationInputFiltering(): Promise<ErrorResponse> {
  // Build Azure content filter with only safe content allowed for hate and violence
  const azureContentSafetyFilter = buildAzureContentSafetyFilter({
    Hate: 'ALLOW_SAFE',
    Violence: 'ALLOW_SAFE'
  });

  // Build Llama guard content filter with categories 'privacy' enabled
  const llamaGuardFilter = buildLlamaGuardFilter('privacy');

  const orchestrationClient = new OrchestrationClient({
    llm,
    templating,
    filtering: {
      input: {
        filters: [azureContentSafetyFilter, llamaGuardFilter]
      }
    }
  });

  try {
    // Trigger the input filters which results in a 400 Bad Request error
    await orchestrationClient.chatCompletion({
      inputParams: { input: 'My social insurance number is ABC123456789.' } // Should be filtered by the Llama guard filter
    });
    throw new Error('Input was not filtered as expected.');
  } catch (error: any) {
    if (error.cause?.status === 400) {
      logger.info('Input was filtered as expected.');
    } else {
      throw error;
    }
    return error.cause?.response?.data;
  }
}

```

Contributors


Stoyanov, Velizar


Antonio, Dan


Marques, Luis

RF
Robledo, Francisco


CIGAINA, MARCO

Thank you