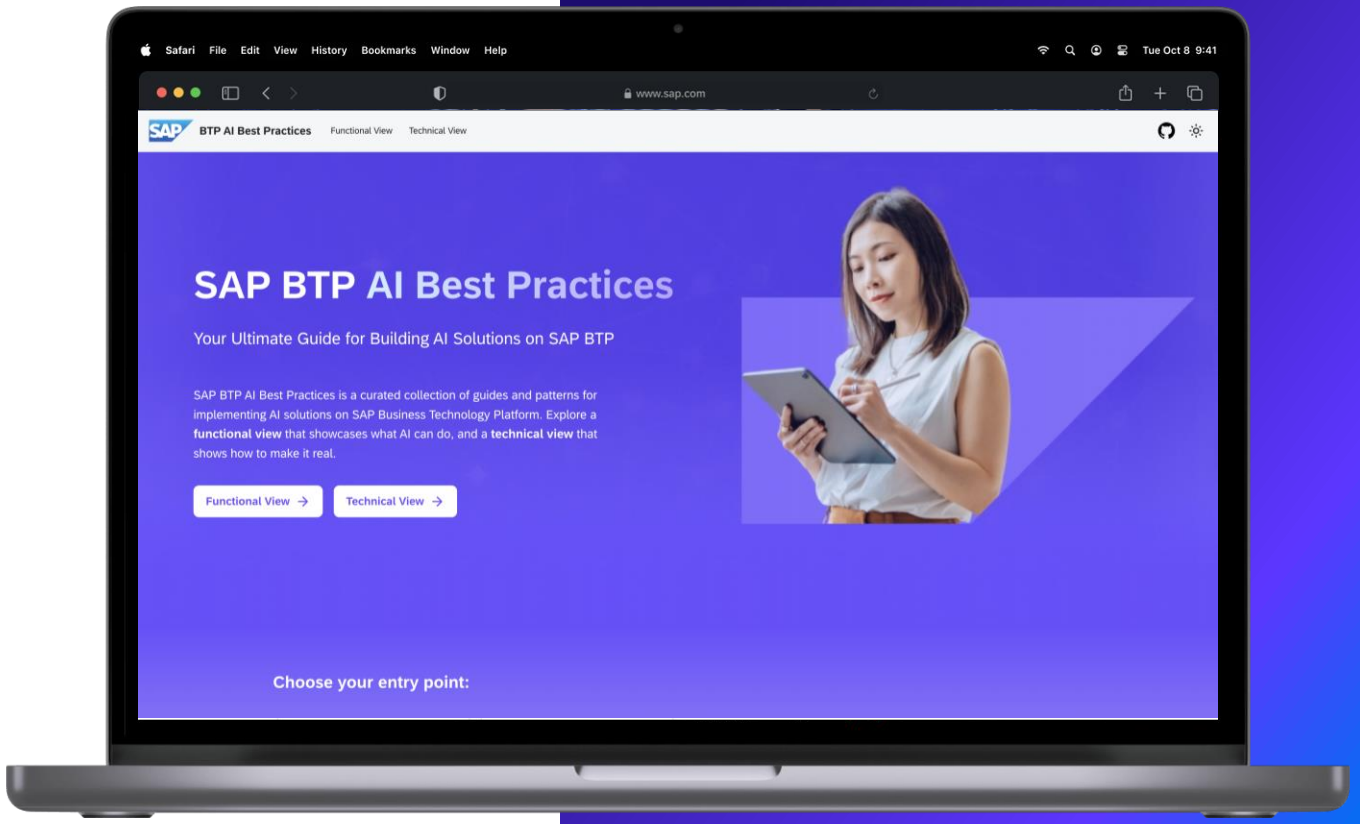
SAP BTP AI Best Practices

Data Masking

A structured approach to efficiently **protect sensitive information** when interacting with Large Language Models.



BTP AI Services Center of Excellence

12.05.2025

Steps

1 Overview

2 Pre-requisites

3 Key Choices and Guidelines

4 Implementation

Data Masking

Simple Masking of Private Identifiable Data for Generative AI Models

Crucial technique used to **protect sensitive information** when interacting with Large Language Models (LLMs)

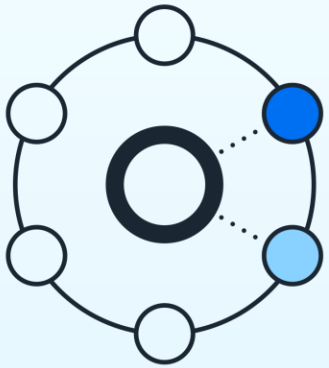
Involves replacing **identifiable** or **confidential** data within prompts with **placeholder text**, ensuring that such information **is not exposed** to third-party models.

Expected Outcome

- Ensure that sensitive information is protected
- Ensure data privacy and security.

Key Benefits

Why use BTP AI Core instead of direct access?



Safe Use in Non-Production Environments

Developers, testers, and analysts can work with realistic datasets without compromising actual customer information. This enables better software development and analytics while maintaining data privacy.

Enhanced Data Security

Data masking protects sensitive information (like PII, PHI, or financial data) by replacing it with realistic but fictional data. This prevents unauthorized access to real data, reducing the risk of data breaches and leaks.

Regulatory Compliance

Helps organizations comply with data privacy laws and standards (e.g., GDPR, HIPAA, PCI-DSS) by ensuring that sensitive data is not exposed in non-production environments like testing, training, or analytics.

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)

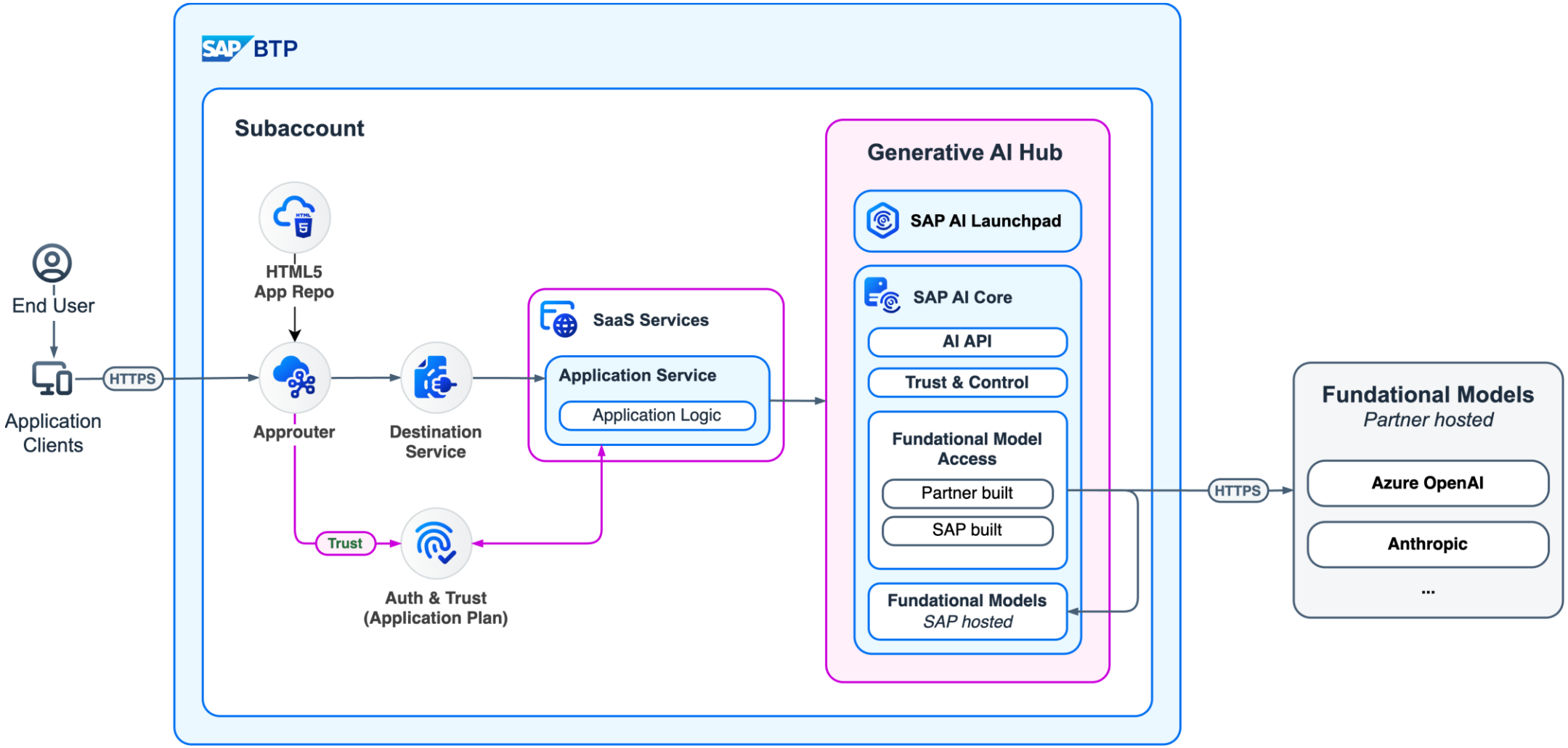
SAP Business Technology Platform (SAP BTP)

- 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

Data Masking

Key Choices and Guidelines

Data Masking

Define the scope of data masking:

Determine which data fields need to be masked based on sensitivity and compliance requirements.

- **Guideline**: Identify sensitive data types to be masked, such as:
  - Personally Identifiable Information (PII): names, emails, phone numbers, IDs
  - Financial data: credit card numbers, account balances
  - Health data (for HIPAA compliance)

- To see the full list of PII detected by **AI Core** anonymization service you can refer to this wiki.

- **Choice**: Use static rules (e.g., regex patterns) or ML-based classifiers for data detection.

Key Choices and Guidelines

Data Masking

Select masking techniques:

Choose appropriate masking techniques such as encryption, tokenization, or pseudonymization.

- **Anonymization**: replaces personally identifiable information in chosen categories with a placeholder (e.g.,{{MASKED_ENTITY}}). This method is irreversible, meaning the original data cannot be retrieved once anonymized. It is useful for scenarios where data privacy is paramount, but it may limit the model's ability to process the input due to loss of context
- **Pseudonymization**: substitutes personally identifiable information with reversible tokens (e.g.,{{MASKED_ENTITY_ID}}). This method allows the original data to be unmasked in the response, providing a balance between privacy and usability
- **Redaction**: Fully remove or obscure the sensitive parts

Current data masking SAP model used by AI Core supports only anonymization for compliance purposes

Key Choices and Guidelines

Data Masking

Preserve Prompt Intent

- **Guideline**: Ensure that masking doesn't distort the semantic meaning of the input. For example:
  - Instead of "John Doe ordered 10 boxes," → use "{{CUSTOMER_NAME}} ordered 10 boxes."
- **Choice**: Balance between anonymization and utility—over-masking may reduce LLM accuracy.

Key Choices and Guidelines

Data Masking

Evaluate Compliance & Risk

- **Guideline**: Ensure your approach aligns with company policies and regulations (e.g., GDPR, HIPAA).
- **Choice**: Work with legal and data privacy teams to define acceptable masking standards.

Key Choices and Guidelines

Data Masking

- **Test the implementation:** Conduct thorough testing to ensure that masked data is correctly processed and retains its utility for AI models.
- **Monitor and audit:** Continuously monitor and audit the masked data to ensure compliance and effectiveness of the masking techniques.

- **Guideline**: Regularly test prompts with masked vs. unmasked inputs to ensure quality and accuracy.
- **Choice**: Include masking logic in your CI pipeline or prompt evaluation tools.

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
- SAP Generative AI hub SDK - Basic Orchestration Pipeline

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

```
async function orchestrationCompletionMasking(): Promise<
  string | undefined
> {
  const orchestrationClient = new OrchestrationClient({
    llm: {
      model_name: 'gpt-4-32k'
    },
    templating: {
      template: [
        {
          role: 'user',
          content:
            'Please write an email to {{?user}} ({{?email}}), informing them about the amazing capabilities of generative AI! Be brief and concise, write at most 6 sentences.'
        }
      ]
    },
    masking: {
      masking_providers: [
        {
          type: 'sap_data_privacy_integration',
          method: 'pseudonymization',
          entities: [{ type: 'profile-email' }, { type: 'profile-person' }]
        }
      ]
    }
  });

  const response = await orchestrationClient.chatCompletion({
    inputParams: { user: 'Alice Anderson', email: 'alice.anderson@sap.com' }
  });
  return response.getContent();
}

```

Contributors


Stoyanov, Velizar


Antonio, Dan


Marques, Luis

RF
Robledo, Francisco


CIGAINA, MARCO

Thank you