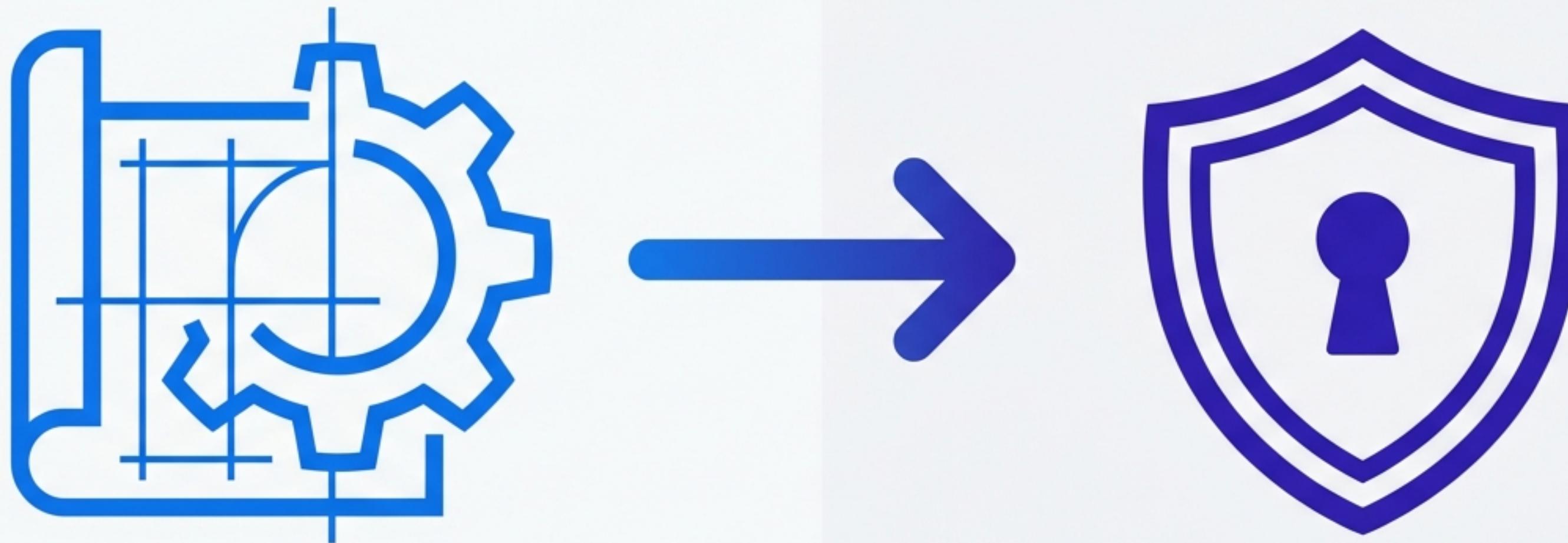


**Da sua Máquina  
para a Nuvem:**

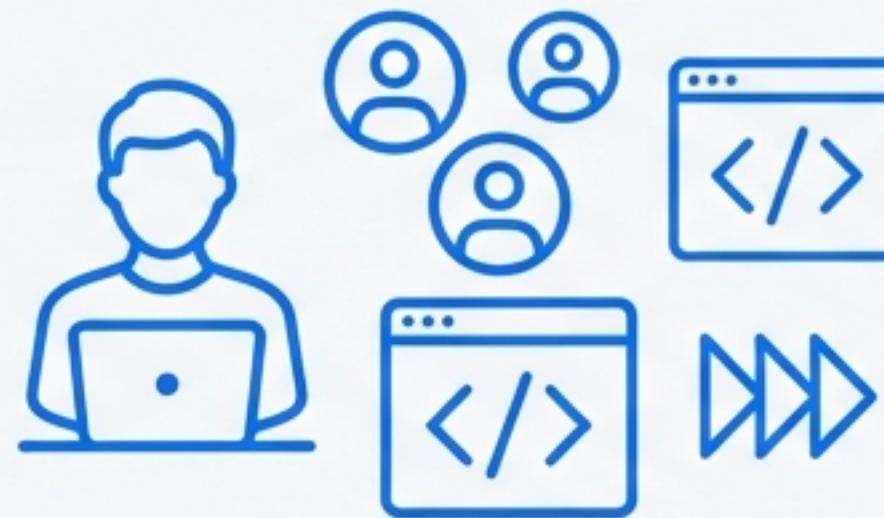
**Configurando Aplicações  
CAP Java para Produção**



Um guia prático para garantir segurança e performance na SAP BTP.

# O Desafio: Gerenciar Configurações Entre Ambientes

## Desenvolvimento Local



Foco em agilidade. Recursos como mock users, páginas de índice e acesso interno são essenciais para testes rápidos.

## Ambiente de Produção

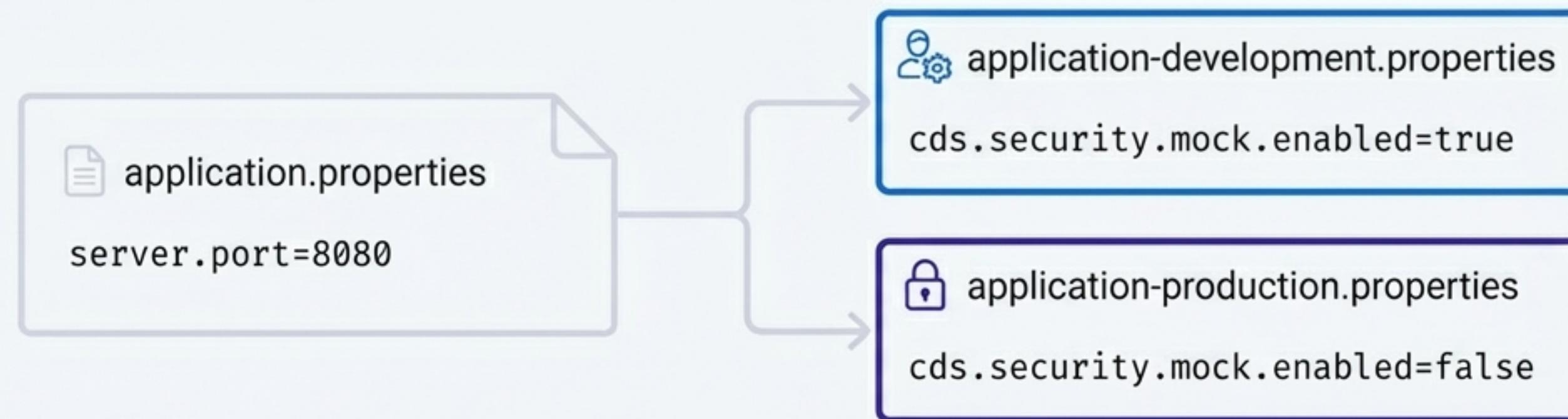


Foco em segurança e estabilidade. As mesmas funcionalidades de desenvolvimento se tornam vulnerabilidades críticas se ativadas em produção.

*Como garantir que as configurações de desenvolvimento nunca cheguem ao ambiente produtivo por acidente?*

# A Estratégia: Isolar Configurações com Spring Profiles

Aplicações CAP Java utilizam todo o poder da *Externalized Configuration* do Spring Boot. Isso permite definir múltiplos perfis de configuração para cenários distintos.



## Como Funciona:

- application.properties: Configurações base.
- application-development.properties: Overrides específicos para desenvolvimento.
- application-production.properties: Overrides específicos para produção.

Esta abordagem é o padrão da indústria para criar aplicações robustas e adaptáveis.

# A Solução CAP Java: O Perfil de Produção Integrado

Para simplificar e padronizar, o CAP Java oferece um perfil especial e pré-configurado chamado `production`.



Ele ativa um conjunto de padrões de configuração recomendados para ambientes de produção com um único comando, garantindo que sua aplicação inicie de forma segura por padrão.

*“O *production profile* configura um conjunto de padrões de propriedade selecionados, recomendados para implantações de produção, de uma só vez.”*

# O Que o Perfil `production` Desativa Automaticamente

Ao ativar este perfil, as seguintes propriedades são configuradas para `false`, reforçando a segurança do seu ambiente.

## 1. Página de Índice Desativada

`cds.index-page.enabled`

Evita a exposição de endpoints e serviços.

## 2. Mock Users Desativados

`cds.security.mock.enabled`

Remove usuários de teste que poderiam ser uma porta de entrada para ataques.

## 3. Acesso de Teste Interno Desativado

`cds.security.authentication.internalUserAccess.enabled`

Fecha o acesso a usuários técnicos internos não destinados a produção.

**Nota Importante:** Configurações explícitas na sua aplicação (ex: em `application.properties`) têm precedência sobre os padrões do perfil de produção.



# ALERTA DE SEGURANÇA

Não habilite manualmente em produção os recursos que são desativados pelo perfil de produção.

Fazer isso pode introduzir sérias vulnerabilidades de segurança na sua aplicação.

# Ação #1: Ativando o Perfil no seu Descriptor de Implantação

**Instrução:** A ativação de perfis em produção é feita através da variável de ambiente `SPRING_PROFILES_ACTIVE`.

**Contexto:** Configure esta variável no seu `mta.yaml` ou outros descritores de implantação (como Helm charts).

```
# Em seu mta.yaml, dentro da definição do seu módulo Java

- name: my-service
  type: java
  path: srv
  parameters:
    buildpack: sap_java_buildpack_jakarta
  # ... outras configurações
  provides:
    - name: srv-api
      properties:
        url: ${default-url}
  requires:
    - name: my-db
  properties:
    # A LINHA ESSENCIAL PARA ATIVAR O PERFIL DE PRODUÇÃO
    SPRING_PROFILES_ACTIVE: cloud
```



**Nota:** Por padrão, o perfil de produção do CAP é o `cloud`. Para usar um perfil customizado, defina a propriedade `cds.environment.production.profile`.

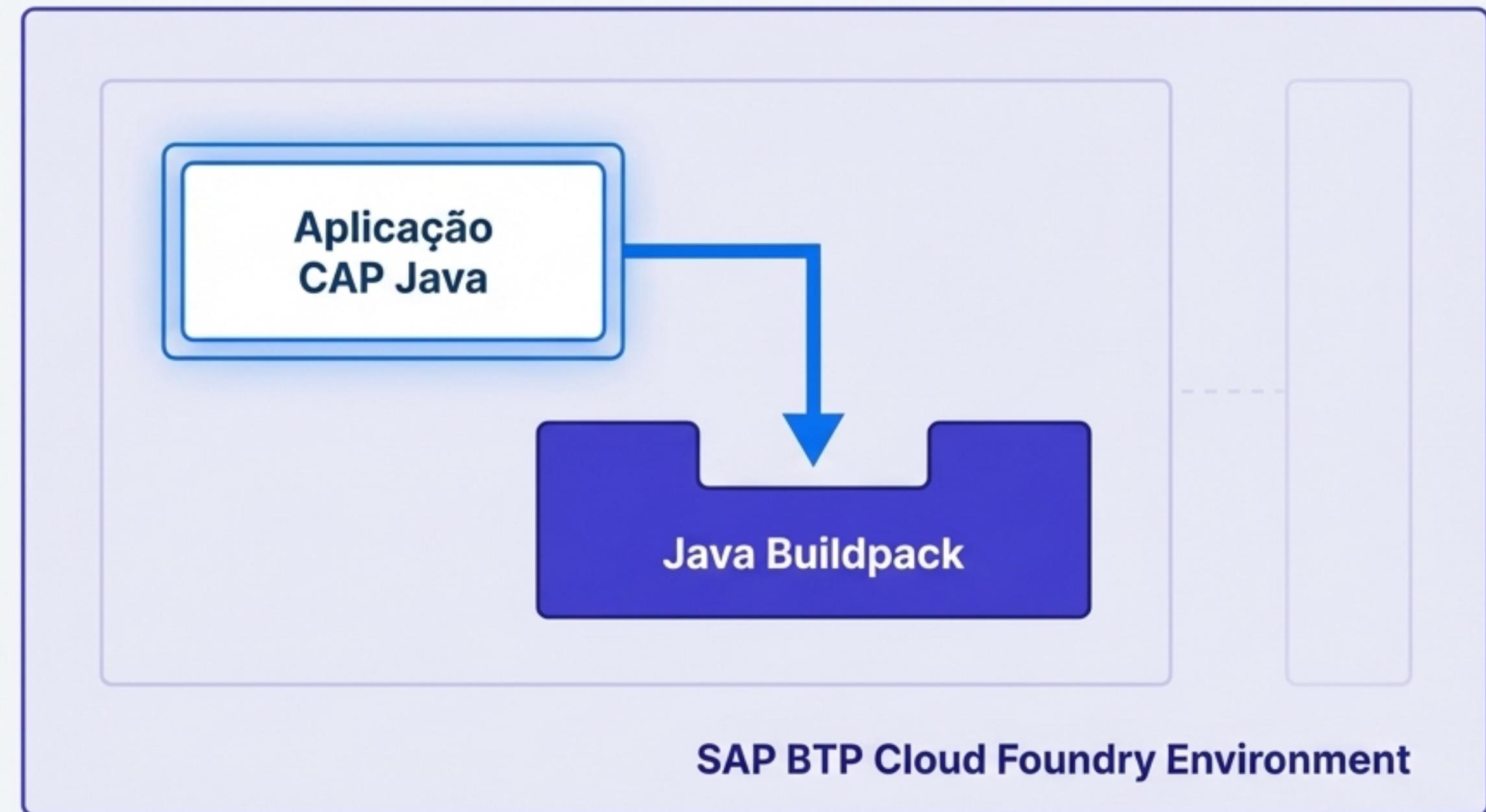
# Da Aplicação para o Ambiente: Configurando o Runtime Java

## Conceito

Uma aplicação segura precisa rodar em um ambiente igualmente seguro e otimizado.

## Introdução

No ambiente SAP BTP Cloud Foundry, o runtime Java que executa sua aplicação é definido por um **buildpack**.



SAP BTP Cloud Foundry Environment

## Ação #2: Especificando o Buildpack e a Versão do Java

Recomendação: Para aplicações CAP, utilize o **SAP Java Buildpack**. Para aplicações Jakarta EE (padrão em versões recentes), a versão correta é `sap\_java\_buildpack\_jakarta`.

Instrução: Adicione as seguintes linhas ao seu `mta.yaml` para usar o buildpack correto e configurar o Java 21 (SapMachine JRE).

```
# Em seu mta.yaml, na seção de parâmetros do seu módulo Java

parameters:
    buildpack: sap_java_buildpack_jakarta    # Define o buildpack recomendado
    # ...
properties:
    JBP_CONFIG_SAP_MACHINE_JRE: '{ version: 21.+ }'  # Força o uso do Java 21+
```

Nota: Aplicações CAP com Spring Boot não requerem configuração específica adicional e rodam usando o runtime Java *Main* por padrão.

# Nota de Compatibilidade: SAP Business Application Studio

## Cenário

O que fazer se o Java 21 não estiver disponível no seu ambiente de desenvolvimento?

## Instrução Clara

Se você desenvolve sua aplicação no SAP Business Application Studio e o Java 21 não está disponível, utilize o Java 17.



## Exemplo de configuração para Java 17

```
# Exemplo de configuração para Java 17  
  
properties:  
  JBP_CONFIG_SAP_MACHINE_JRE: '{ version: 17.+ }'
```

# Sua Checklist Essencial para Produção

Para garantir que sua aplicação CAP Java seja segura e robusta em produção, confirme sempre estes dois pontos críticos no seu `mta.yaml`:



## 1. Ativar o Perfil de Produção:

Adicione a propriedade `SPRING\_PROFILES\_ACTIVE: cloud` para habilitar os padrões de segurança.



## 2. Configurar o Buildpack Correto:

Defina o `buildpack: sap\_java\_buildpack\_jakarta` e a versão do Java via `JBP\_CONFIG\_SAP\_MACHINE\_JRE`.

Configurações corretas não são opcionais. Elas são a base de uma aplicação profissional e segura na SAP BTP.