

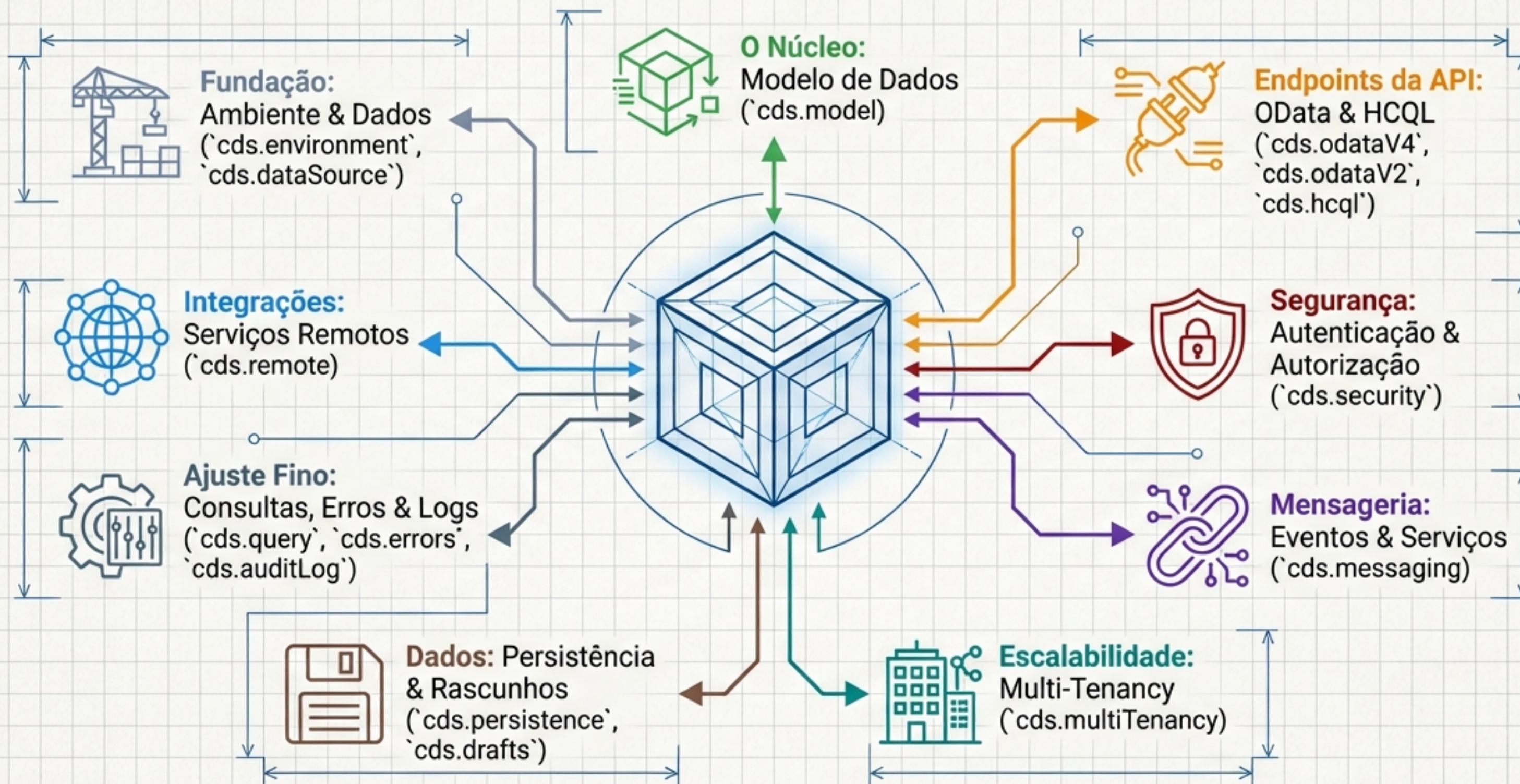
O Blueprint da sua Aplicação CAP Java

Um Guia Arquitetural para a Configuração
Completa do Framework

Baseado nas propriedades do CAP Java 4.5.0.

O Mapa da Arquitetura: Uma Visão Guiada da Configuração

As propriedades de configuração do CAP Java não são apenas uma lista; elas formam o blueprint da sua aplicação. Vamos explorar cada componente principal, desde a fundação até os serviços de ponta.





Fundação: Configurando o Ambiente e a Fonte de Dados

Define o contexto de execução (local, produção) e como sua aplicação se conecta ao banco de dados. Esta é a base para o funcionamento de todo o sistema.

Configuração Detalhada

Propriedade	Tipo	Descrição
<code>cds.environment.local.defaultEnvPath</code>	String	Caminho para o arquivo `*.env` padrão no desenvolvimento local.
<code>cds.dataSource.csv.initializationMode</code>	String	Modo de inicialização de dados a partir de arquivos CSV (`always`, `never`).
<code>cds.dataSource.csv.paths</code>	List<String>	Caminhos para os diretórios contendo os arquivos CSV.
<code>cds.dataSource.autoConfig.enabled</code>	boolean	Habilita ou desabilita a configuração automática do data source.

Os Controles Principais

`cds.environment.production.profile`

Define o perfil Spring a ser ativado em ambiente de produção (ex: `cloud`).

`cds.dataSource.embedded`

Use `true` para um banco de dados H2 em memória, ideal para desenvolvimento local e testes rápidos.

`cds.dataSource.binding`

Especifique o nome do binding do serviço de banco de dados no ambiente de nuvem.

O Núcleo: Definindo e Carregando seu Modelo de Dados

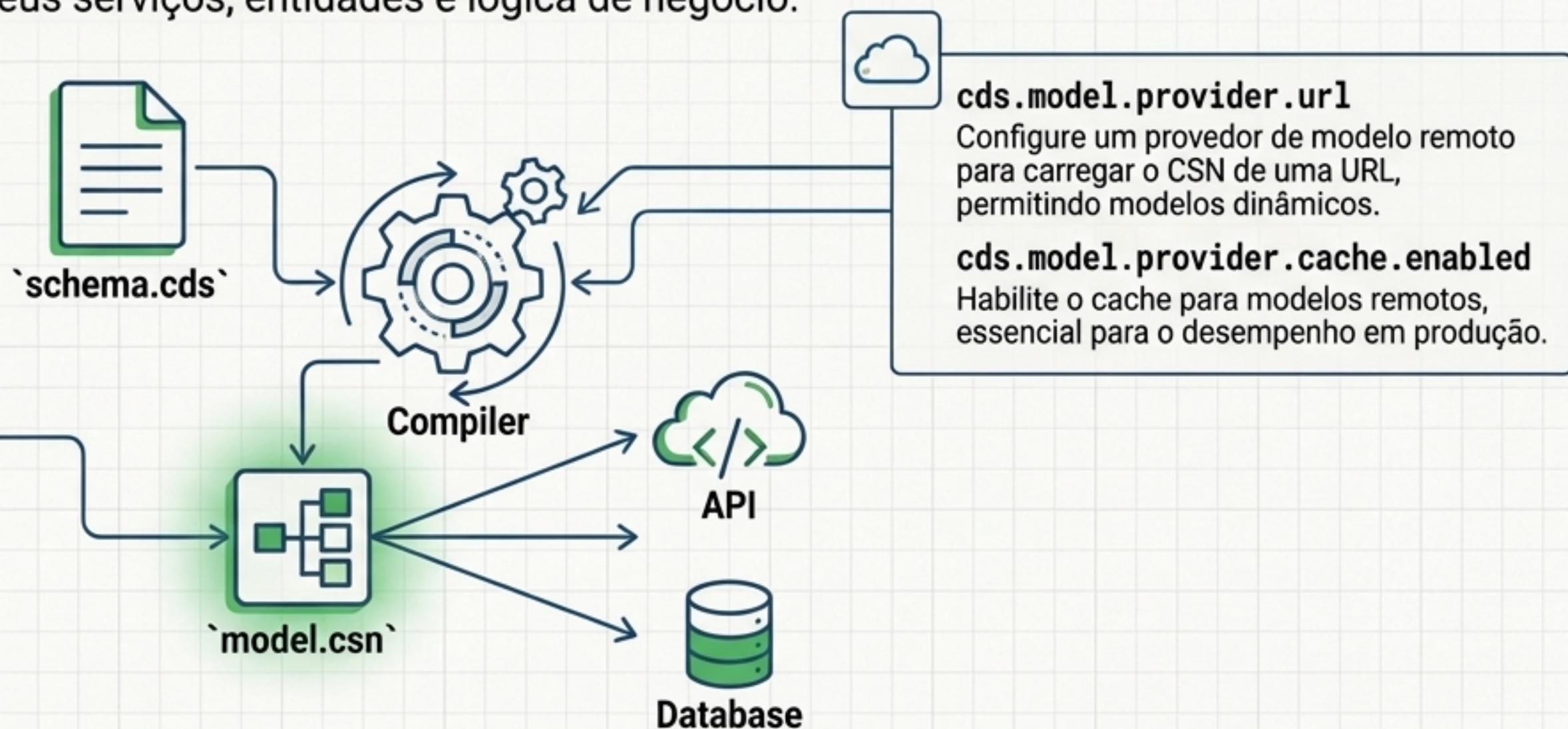
Controla como o framework localiza, carrega e interpreta seus arquivos de modelo CDS (CSN).

O modelo é a fonte da verdade para seus serviços, entidades e lógica de negócio.

Os Controles Principais

`cds.model.csnPath`

Especifique o caminho exato para seu arquivo CSN compilado, otimizando o tempo de inicialização.



Configuração Detalhada

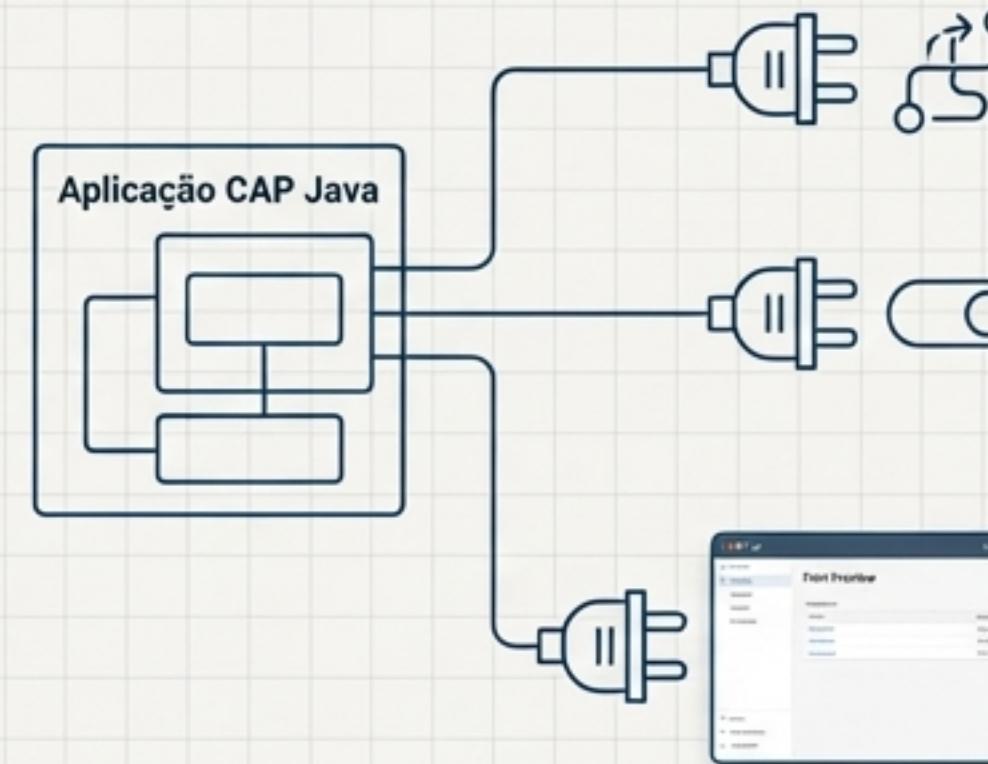
Propriedade	Tipo	Descrição
<code>cds.model.includeUiAnnotations</code>	boolean	Inclui automaticamente anotações de UI no modelo carregado.
<code>cds.model.provider.extensibility</code>	boolean	Habilita o carregamento de extensões de modelo do provedor.
<code>cds.model.provider.cache.maxSize</code>	int	Número máximo de modelos CSN cacheados.
<code>cds.model.provider.cache.expirationTime</code>	int	Tempo de expiração (em segundos) para o cache do modelo.



A Fachada: Expondo sua Aplicação com Endpoints OData

Gerencia como os serviços definidos no seu modelo são expostos como endpoints OData V4 e V2, controlando seus caminhos, formatos e comportamentos.

Os Controles Principais



cds.odataV4.endpoint.path
Define o caminho base para todos os serviços OData V4 (padrão: /odata/v4).

cds.odataV2.endpoint.enabled
Habilite true para expor uma versão OData V2 do seu serviço para compatibilidade com clientes mais antigos.

cds.odataV4.fioriPreview.enabled
Ative para obter uma página de pré-visualização Fiori Elements durante o desenvolvimento local, acelerando o teste de UI.

Configuração Detalhada

Propriedade	Tipo	Descrição
cds.odataV4.edmxPath	String	Caminho customizado para o documento de metadados EDMX.
cds.odataV4.searchMode	String	Modo de busca a ser utilizado (highlight` ou `phrase`).
cds.odataV4.batch.maxRequests	long	Número máximo de requisições permitidas em uma única chamada \$batch.
cds.hcql.endpoint.path	String	Define o caminho para o endpoint do CAP Java HCQL (Human-Readable CQL).

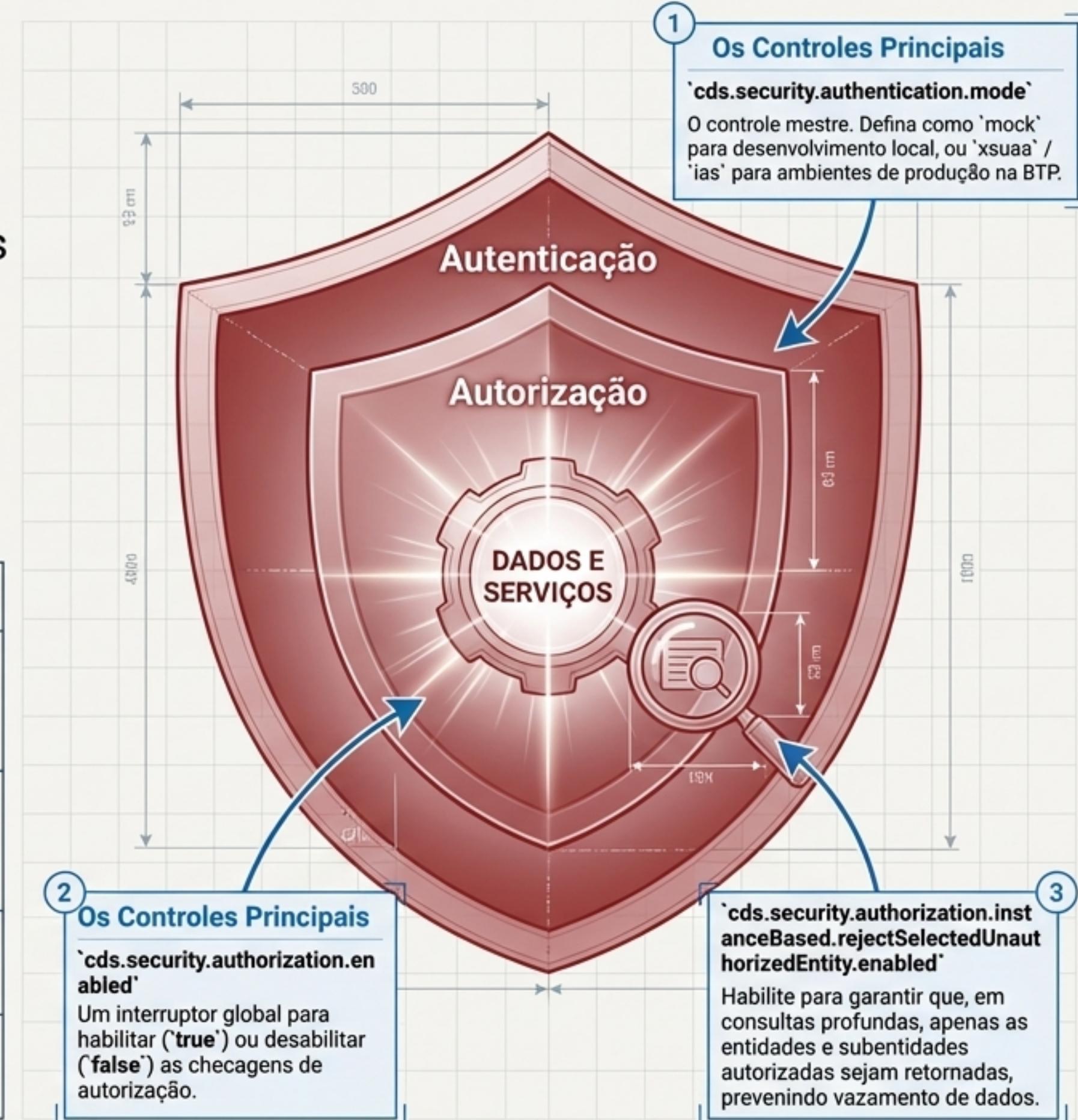


A Estrutura de Segurança: Autenticação e Autorização

Define como sua aplicação verifica a identidade dos usuários (autenticação) e aplica as permissões definidas no seu modelo (autorização) para proteger o acesso aos dados e serviços.

Configuração Detalhada

Propriedade	Tipo	Descrição
cds.security.logPotentiallySensitive	boolean	Permite o log de informações sensíveis (tokens, atributos) para depuração.
cds.security.authentication.authenticateMetadataEndpoints	boolean	Exige autenticação para acessar os endpoints de metadados ('\$metadata').
cds.security.identity.binding	String	Nome do binding do serviço de identidade (IAS).
cds.security.xsuaa.binding	String	Nome do binding do serviço XSUAA.





Dica Pro: Acelerando o Desenvolvimento com Usuários Mock



Para desenvolvimento e testes locais, você não precisa de um provedor de identidade real. A configuração `cds.security.mock` permite simular um ambiente de segurança completo, com múltiplos usuários, tenants e papéis, diretamente no seu `application.yml`.

Exemplo de Configuração de Usuário Mock (`application.yml`)

```
cds:  
  security:  
    authentication:  
      mode: mock  
    mock:  
      enabled: true  
      users:  
        alice:  
          id: alice  
          password: 123  
          roles:  
            - admin  
            - reviewer  
          attributes:  
            Country: ['DE']  
        bob:  
          id: bob  
          password: 123  
          roles:  
            - viewer
```

FIG. 03.1: MOCK_USER_CONFIG.yml

Propriedades Essenciais para Usuários Mock

- id / name: Identificador do usuário.
- password: Senha para autenticação básica.
- tenant: O tenant ao qual o usuário pertence.
- roles: Uma lista de papéis (roles) para testar as autorizações.
- attributes: Um mapa de atributos para testar a segurança baseada em atributos.



Arquitetura Orientada a Eventos: Configurando a Mensageria

Permite que sua aplicação envie e receba mensagens de forma assíncrona, integrando-se a serviços de mensageria como o SAP Event Mesh. Configure os serviços, as rotas e o comportamento do outbox/inbox.

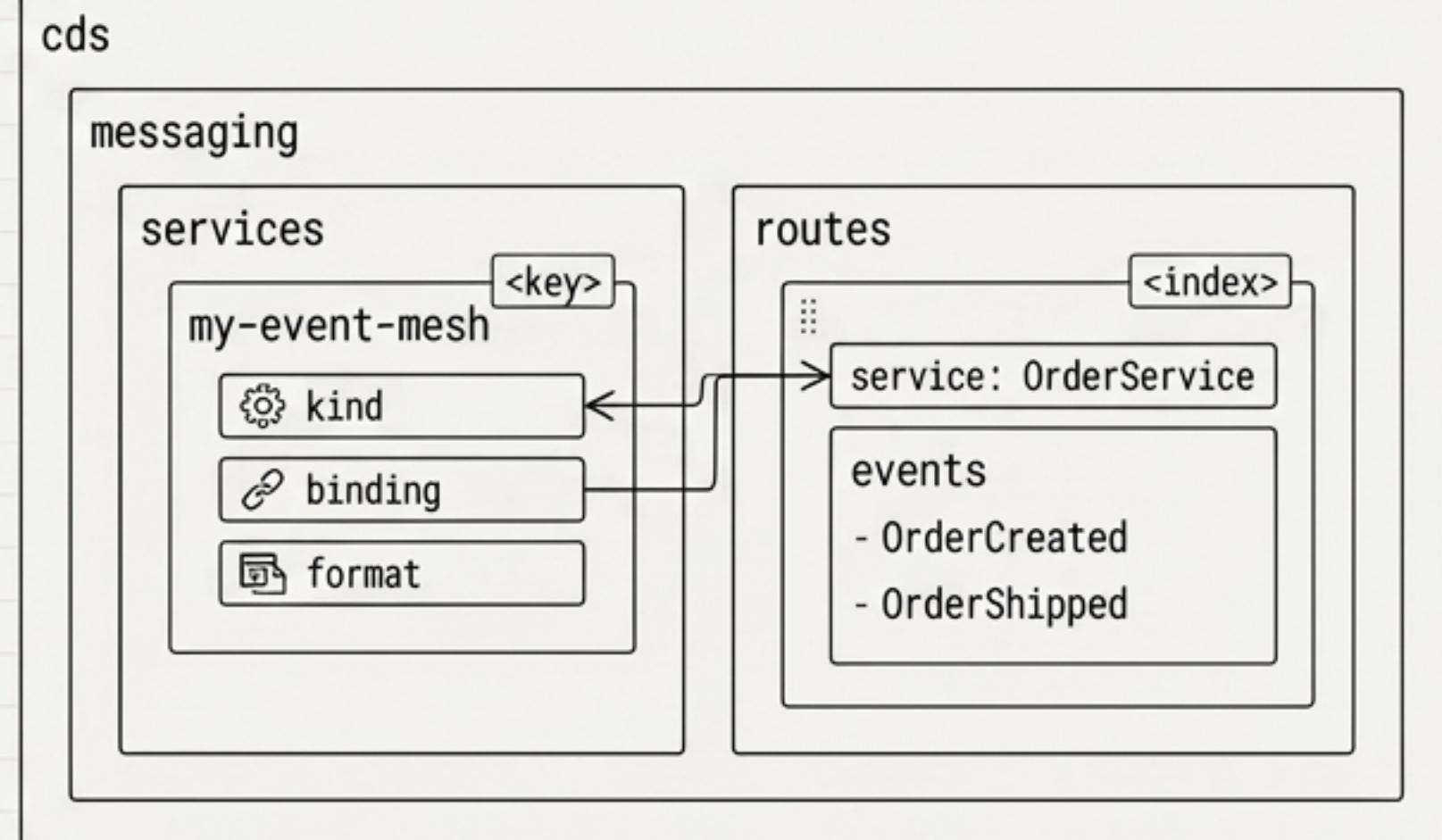
Os Controles Principais

`cds.messaging.services.<key>.kind`: O tipo de serviço de mensageria (ex: `enterprise-messaging-shared`). Este é o principal seletor da implementação.

`cds.messaging.services.<key>.binding`: O nome do binding do serviço na BTP.

`cds.messaging.routes.<index>.service`: Direciona eventos de um serviço CDS (<index>) para um serviço de mensageria específico (<key>).

Estrutura de Configuração (Exemplo Conceitual)

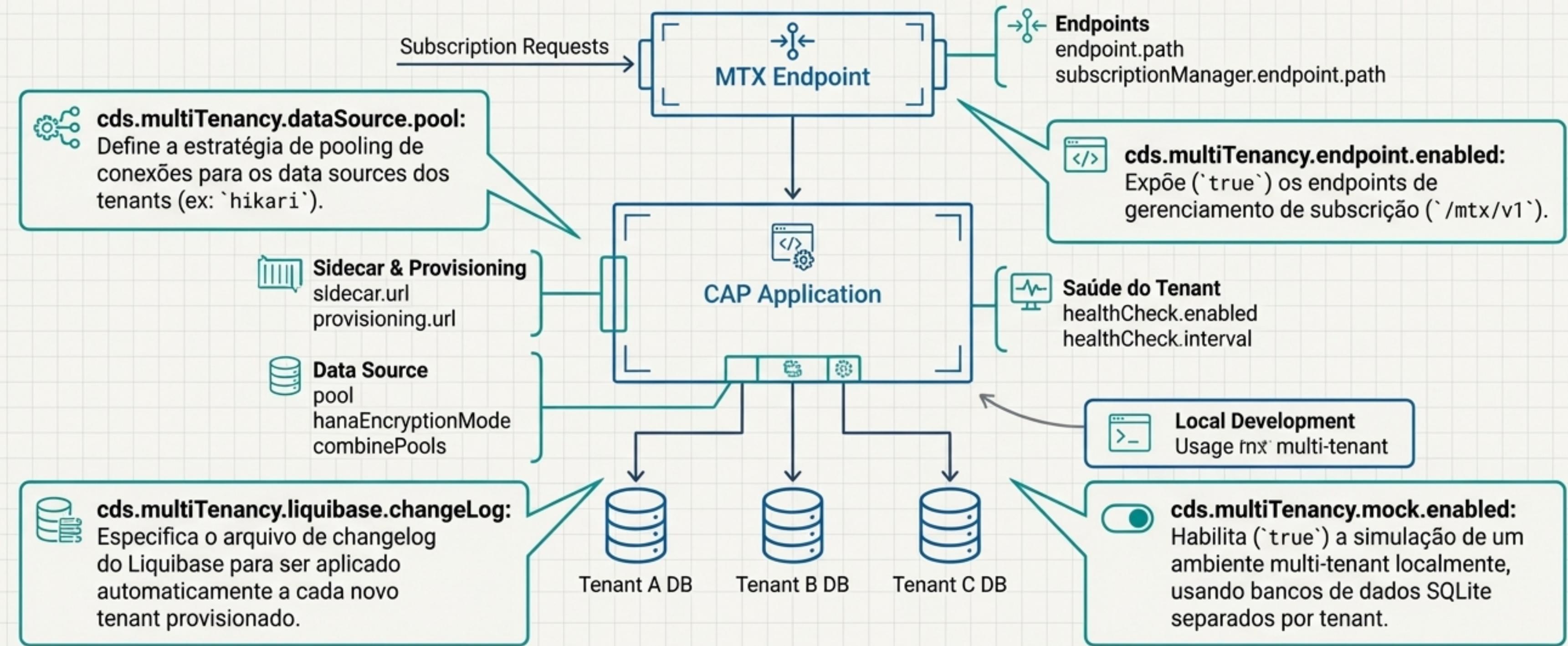


Configurações Importantes de Serviço: `subscribePrefix`, `publishPrefix`, `outbox.enabled`, `inbox.enabled`.



Construindo para Escala: O Blueprint da Multi-Tenancy

Habilita sua aplicação para servir **múltiplos tenants** (clientes) de forma **isolada e segura**. As configurações gerenciam o provisionamento de tenants, o isolamento do banco de dados e a integração com serviços de subscrição.





Garantindo a Integridade: Persistência e Gerenciamento de Rascunhos

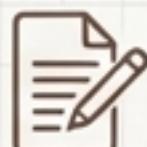
Controla como as alterações de dados são escritas no banco de dados e como os rascunhos (drafts) são gerenciados.

Essas configurações são vitais para a consistência transacional e para habilitar fluxos de edição seguros.

Os Controles Principais



`cds.persistence.changeSet.enforceTransactional`:
Quando `true`, garante que todas as operações dentro de um changeset sejam executadas em uma única transação atômica.



`cds.drafts.persistence`: Define o mecanismo de persistência para os rascunhos (ex: `database-managed`).



`cds.drafts.enforceReadonly`: Protege campos marcados como `readonly` contra modificações em rascunhos, aumentando a segurança do modelo.

Configuração Detalhada

Propriedade	Tipo	Descrição
cds.drafts.cancellationTimeout	Duration	Tempo limite para o cancelamento de um rascunho.
cds.drafts.gc.enabled	boolean	Habilita o Garbage Collector para limpar rascunhos órfãos.
cds.drafts.gc.interval	Duration	Intervalo de execução do Garbage Collector.
cds.persistence.services.<key>.dataSource	String	Associa um serviço de persistência a um data source específico.

Ajuste Fino: Otimizando Consultas e Comportamento SQL

Oferece controle granular sobre a execução de consultas, como limites de paginação, ordenação implícita e comportamentos específicos do dialeto SQL (especialmente HANA), permitindo otimizações de performance e segurança.

The image displays three configuration cards side-by-side:

- Left Card:** Configuration for `cds.query.limit.default`. It features a slider from 0 to MAX, currently set to 0, with a "var" button next to it. A note below states: "Define o número padrão de registros retornados se '\$top' não for especificado, prevenindo consultas excessivamente grandes."
- Middle Card:** Configuration for `cds.query.limit.max`. It features a gauge meter with a warning icon (!) above it. A note below states: "Impõe um limite máximo para o valor de '\$top', protegendo a aplicação contra abuso de recursos."
- Right Card:** Configuration for `cds.sql.hana.search.fuzzy`. It features a toggle switch between "Fuzzy" (selected) and "Exact". Below it is a "Search Behavior" section with a note: "Habilita (true) buscas fuzzy ('CONTAINS') em vez de buscas exatas ('LIKE') ao usar a função 'search()' no HANA."

Configuração Detalhada

Propriedade	Tipo	Descrição
cds.query.limit.reliablePaging.enabled	boolean	Garante uma paginação estável mesmo quando os dados subjacentes mudam.
cds.query.restrictions.expand.maxLevels	int	Limita a profundidade máxima de aninhamento para a cláusula '\$expand'.
cds.sql.maxBatchSize	int	Tamanho máximo do lote para operações de escrita no banco de dados.
cds.sql.inlineCount.mode	String	Controla como a contagem de registros (\$count=true) é calculada.

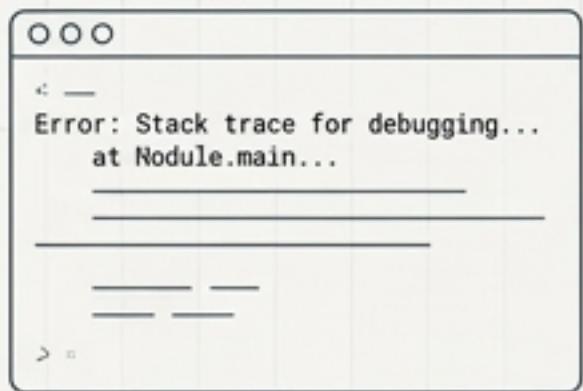
Operação e Conformidade: Erros Detalhados e Log de Auditoria

Configura como a aplicação reporta erros e como registra eventos de auditoria. Essencial para depuração, monitoramento e para atender a requisitos de conformidade e segurança.

Configuração Detalhada

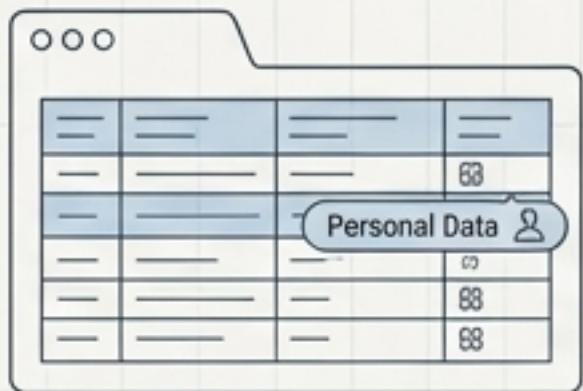
Propriedade	Tipo	Descrição
<code>cds.errors.stackMessages.enabled</code>	boolean	Inclui a cadeia de mensagens de exceção na resposta de erro.
<code>cds.auditLog.v2.enabled</code>	boolean	Habilita o log de auditoria no formato V2.
<code>cds.auditLog.outbox.enabled</code>	boolean	Usa o padrão outbox para o envio de logs de auditoria, aumentando a resiliência.
<code>cds.auditLog.personalData.throwOnMissingDataSubject</code>	boolean	Lança uma exceção se uma entidade de dados pessoais for usada sem um sujeito de dados.

Os Controles Principais



`cds.errors.extended`

Quando `true`, as respostas de erro incluem detalhes técnicos e stack traces, útil para depuração em ambientes de não produção.



`cds.auditLog.personalData.enabled`

Habilita o log de auditoria específico para acesso e modificação de dados marcados como `@PersonalData.IsDataSubject`.



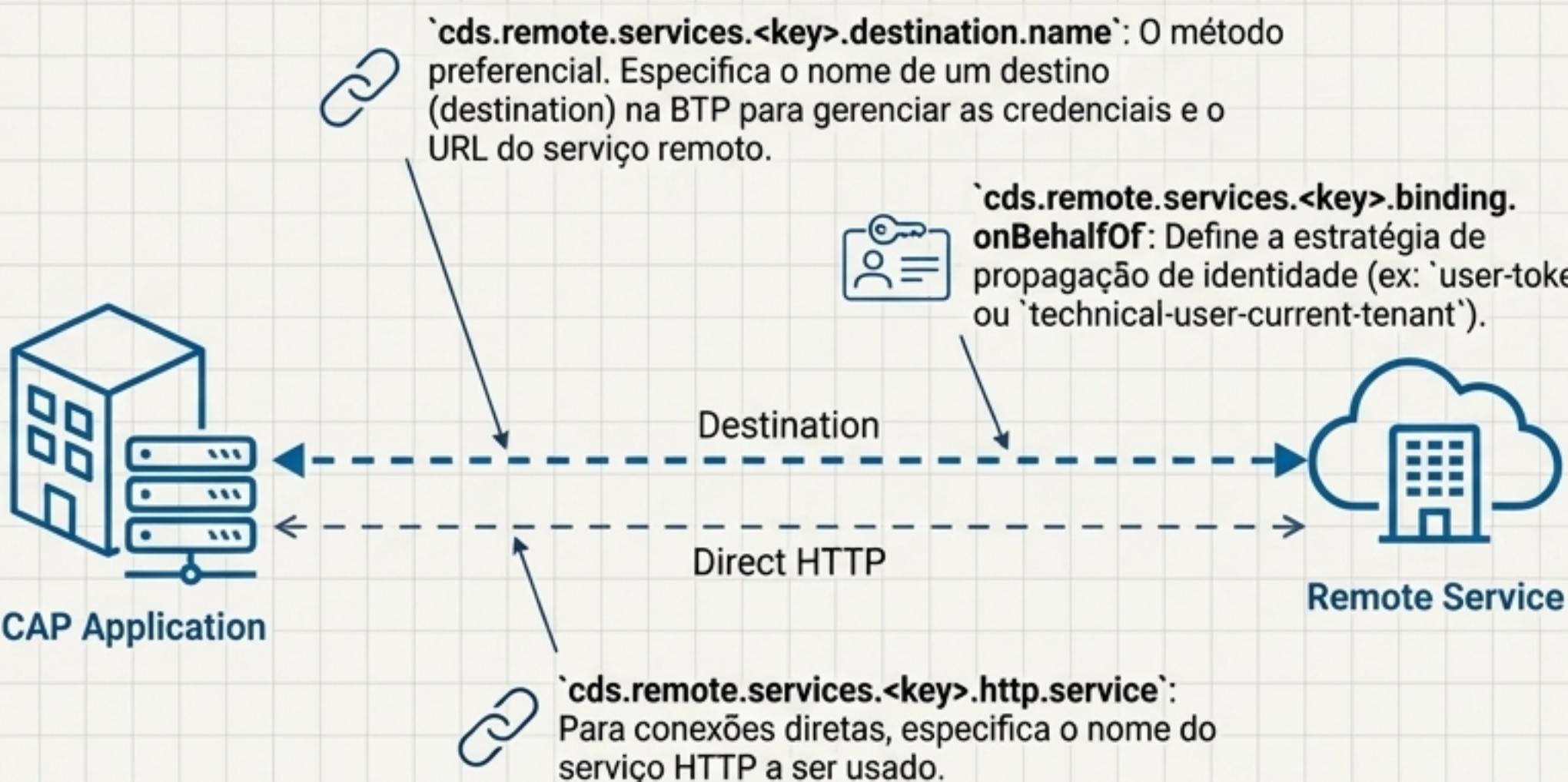
`cds.auditLog.personalData.logRead.enabled`

Ativa o log de auditoria para operações de leitura de dados pessoais, um requisito comum de conformidade.



Além das Fronteiras: Conectando a Serviços Remotos

Core Concept: Define como sua aplicação se conecta e interage com serviços externos (remotos), sejam eles outros microserviços, APIs de parceiros ou sistemas legados. A configuração abstrai os detalhes da conexão, como destinos e autenticação.



**Estrutura de Configuração (Exemplo Conceitual)

```
cds:  
  remote:  
    services:  
      BusinessPartnerAPI: # <key>  
        type: http  
        destination:  
          name: s4-business-partner  
          tokenExchangeStrategy: forwardUserToken
```

Propriedades Adicionais: `model` (caminho para o EDMX do serviço), `http.headers`, `http.csrf.enabled`.



Seu Blueprint Completo: A Arquitetura em Código

A configuração de uma aplicação CAP Java é a sua arquitetura expressa em código. Ao dominar estas propriedades, você não está apenas definindo parâmetros; você está projetando um sistema resiliente, seguro e escalável.

✓ **Pense em Componentes, não em Listas:** Agrupe as configurações pelas áreas arquiteturais que elas controlam.

✓ **Configure por Contexto:** Use perfis para separar as configurações de desenvolvimento, teste e produção.

✓ **A Segurança é a Base:** Configure a autenticação e autorização desde o início, usando mocks para agilidade.

✓ **Domine a Conectividade:** Use as configurações de `messaging` e `remote` para construir sistemas distribuídos robustos.