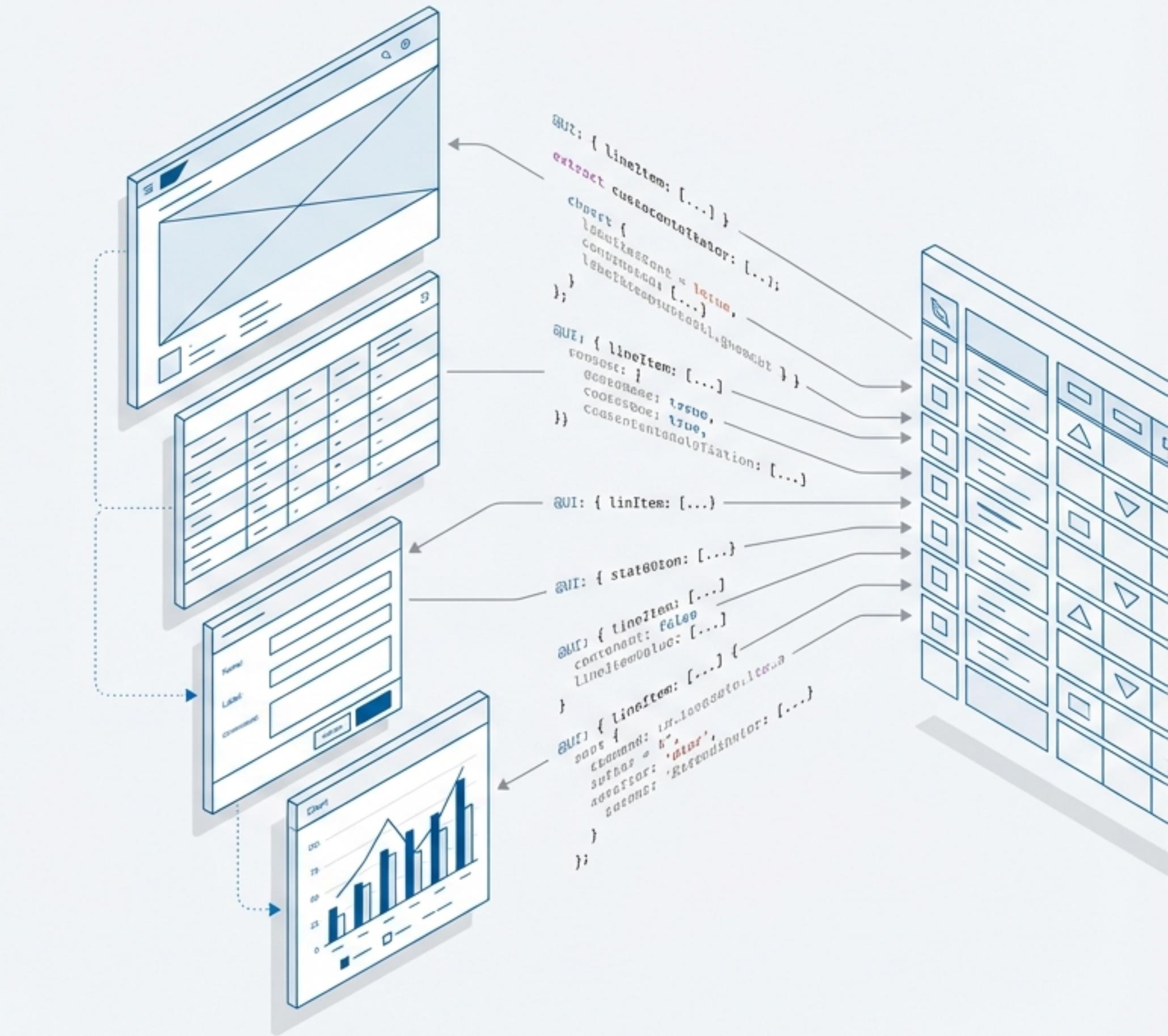


Dominando SAP Fiori Elements com RAP

A Jornada Completa do Desenvolvedor: Da Ideia à Aplicação em Produção



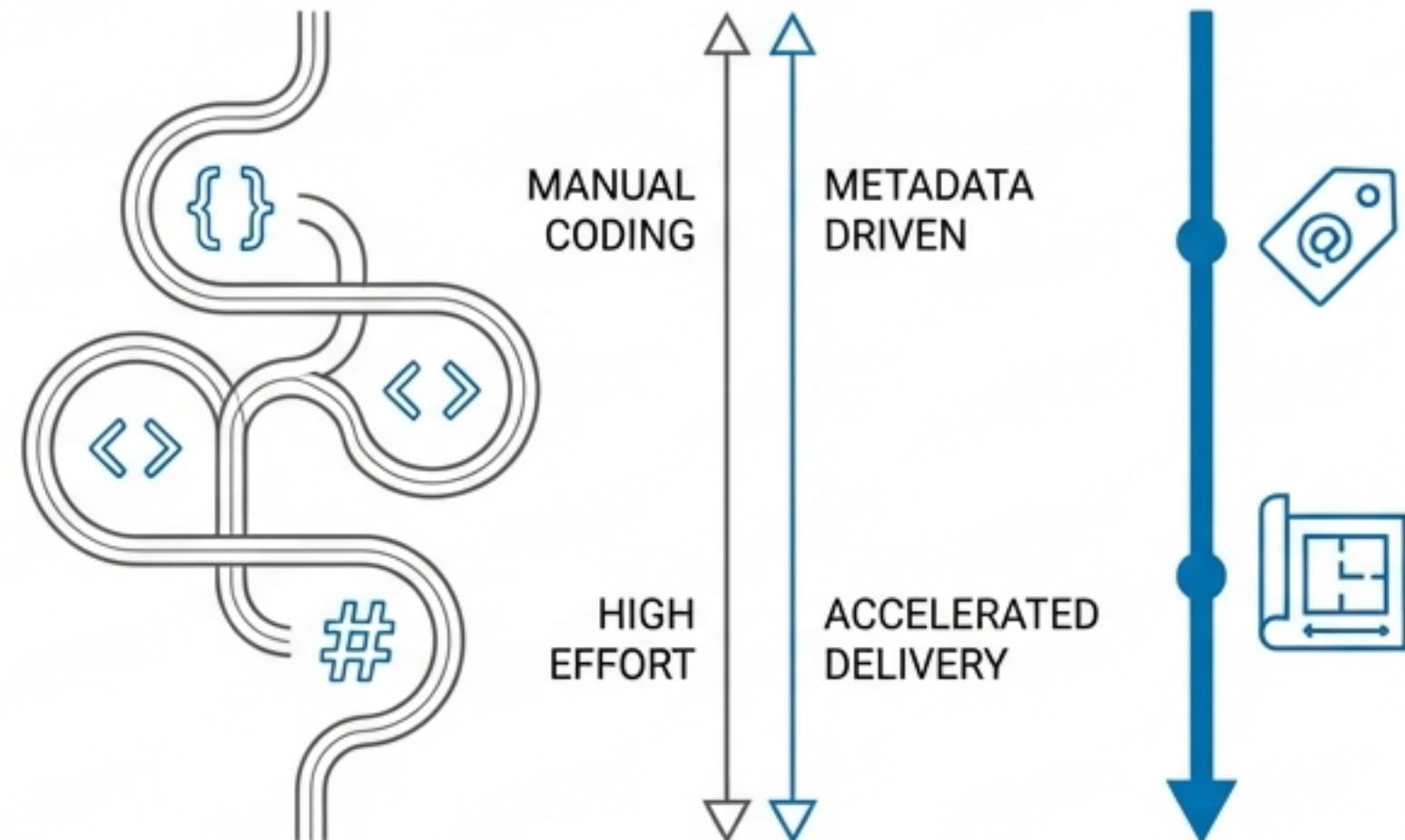
De Artesão de Pixels a Arquiteto de Dados

Freestyle SAPUI5 (O Artesão)

Flexibilidade total, onde cada botão e tabela são codificados manualmente.

Contras

Custo de desenvolvimento e manutenção elevados (alto TCO), lento para entregar, propenso a inconsistências visuais.



SAP Fiori Elements (O Arquiteto)

Biblioteca de templates inteligentes que renderizam a UI com base em metadados do backend.

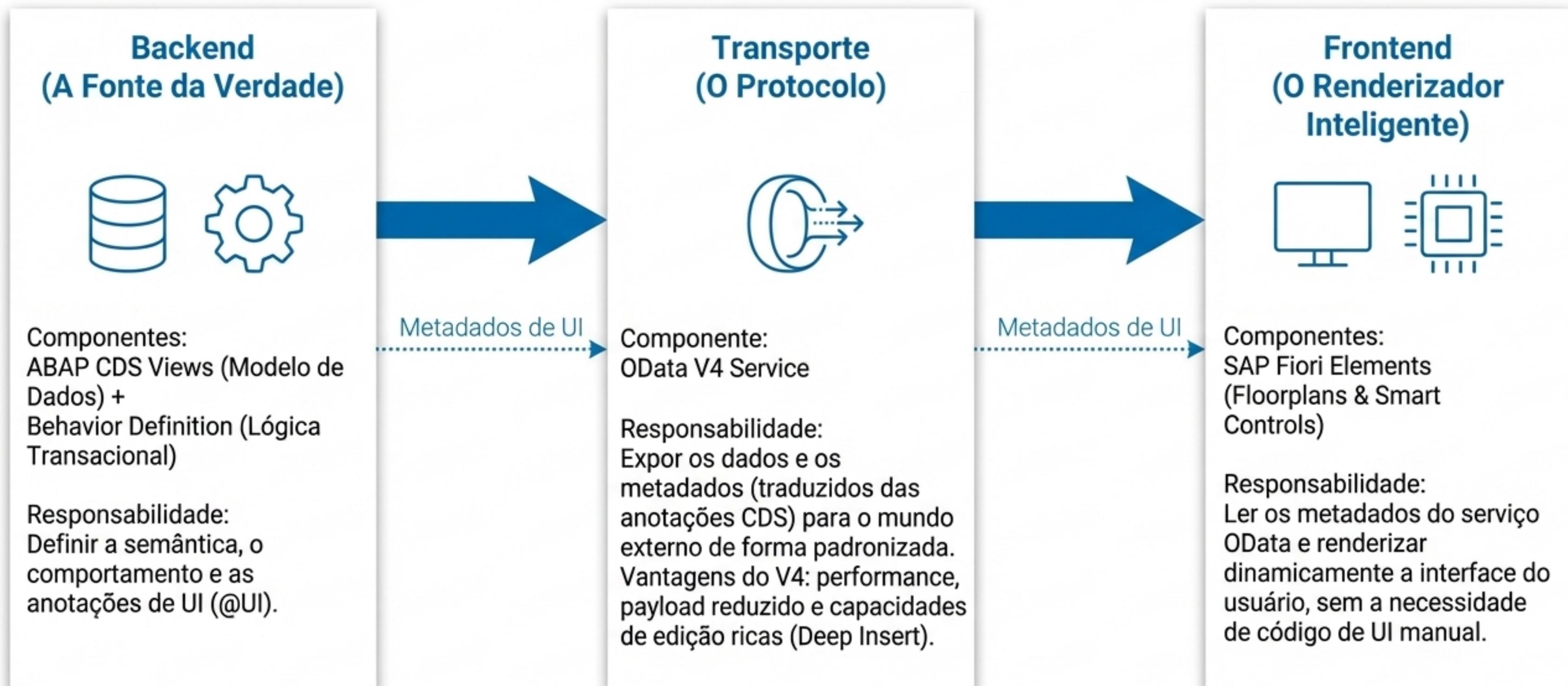
Prós

Desenvolvimento extremamente rápido, baixo TCO, consistência de UX garantida e conformidade automática com o SAP Fiori Design Guidelines.

A Analogia do Lego

Em vez de moldar argila (Freestyle), você recebe blocos prontos (Elements). Você não codifica 'Crie um botão azul aqui'. Você anota no backend: 'Esta entidade é editável'. O Fiori Elements lê a intenção e desenha a tela correta.

A Arquitetura em Uma Página: O Fluxo da Informação



Definindo o Contrato: O "Cardápio" da sua Aplicação



1. Service Definition (O "Cardápio")

Atua como uma Whitelist para selecionar explicitamente quais entidades (CDS Views) farão parte do serviço.

Define o escopo do negócio, garantindo segurança e organização.

```
expose Z_C_TRAVEL as Travel
```

para Aliasing, desacoplando o nome técnico interno do nome público da API.



2. Service Binding (O "Protocolo e Tomada")

"Amarra" uma Service Definition a um endpoint HTTP real e a um protocolo.

- **Protocolo:** OData V4 (padrão) vs. OData V2 (compatibilidade).
- **Tipo de Uso:** **UI** (inclui anotações de UI para Fiori) vs. **Web API** (remove anotações de UI para integrações A2A/B2B).

Descrevendo a Interface, Não a Codificando

Anotações ABAP CDS são a linguagem que o backend usa para instruir o frontend sobre como se desenhar. Você declara intenções, não implementa pixels.

The screenshot illustrates the mapping between ABAP CDS annotations and the resulting Fiori interface. The annotations on the left define the structure and behavior of the UI elements shown on the right.

```
annotate view Z_C_TRAVEL with {
    @UI.lineItem: [ { position: 10 } ]
    @UI.selectionField: [ { position: 10 } ]
    TravelID;
    @UI.headerInfo: {
        typeName: 'Viagem',
        title: { value: 'Description' }
    }
    Description;
    @UI.identification: [ { position: 20 } ]
    AgencyID;
}
```

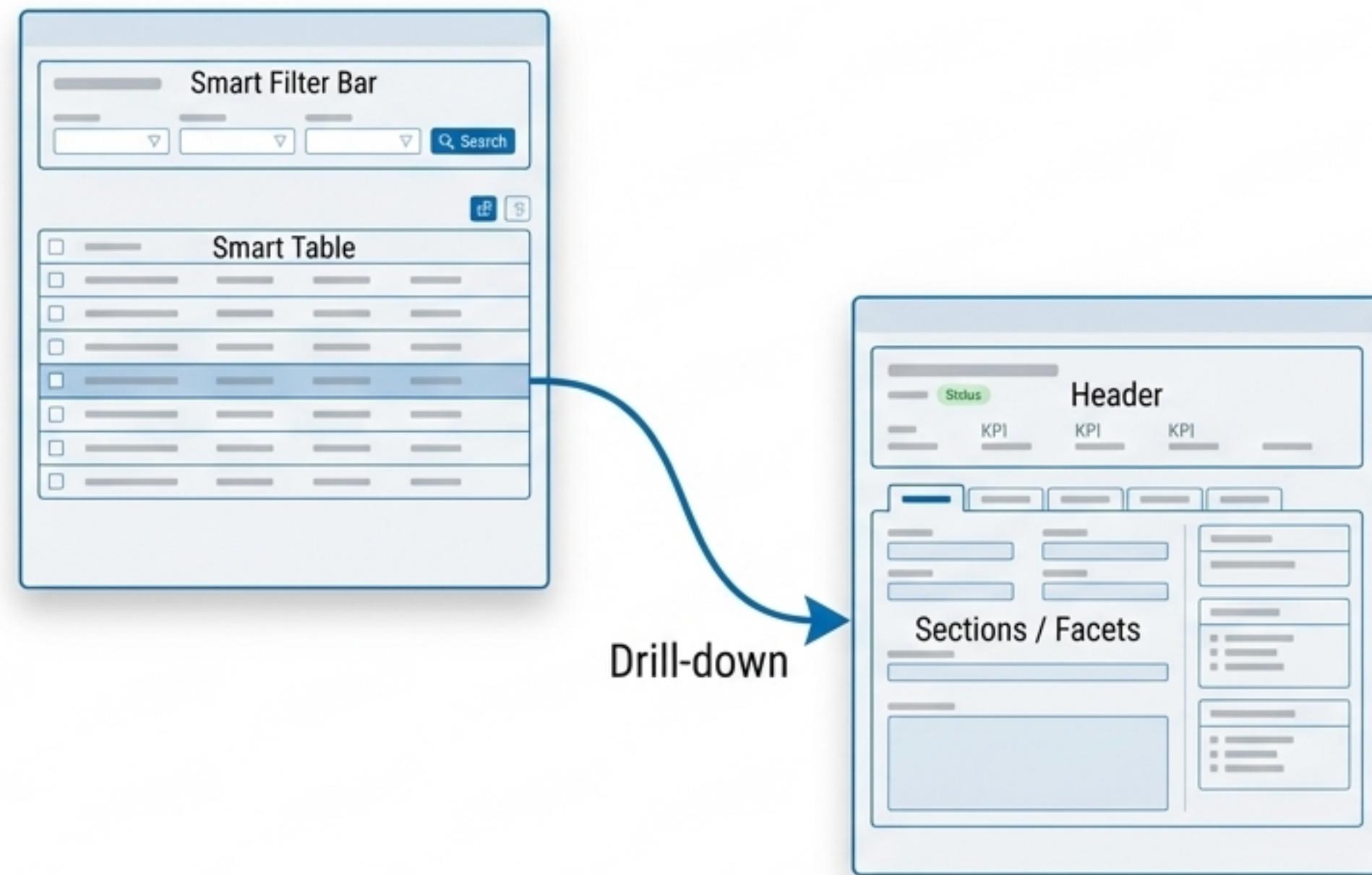
- TravelID:** Annotated with `@UI.lineItem: [{ position: 10 }]`. It points to the first input field in the search bar.
- Description:** Annotated with `@UI.headerInfo: { typeName: 'Viagem', title: { value: 'Description' } }`. It points to the header section of the Object Page, which displays the title "Viagem: [Description Value]".
- AgencyID:** Annotated with `@UI.identification: [{ position: 20 }]`. It points to the "AgencyID:" input field in the "Form" section of the Object Page.

A Jornada do Usuário: Da Lista ao Detalhe

1. List Report

Focus: Encontrar registros em grandes volumes de dados.

- Smart Filter Bar: Área de filtros
- Smart Table: Tabela de resultados
- Ações na Toolbar: Criar, Excluir, Exportar



2. Object Page

Focus: Entender e Modificar um único registro.

- Header: Área de título e KPIs
- Sections / Facets: Corpo da página dividido em abas

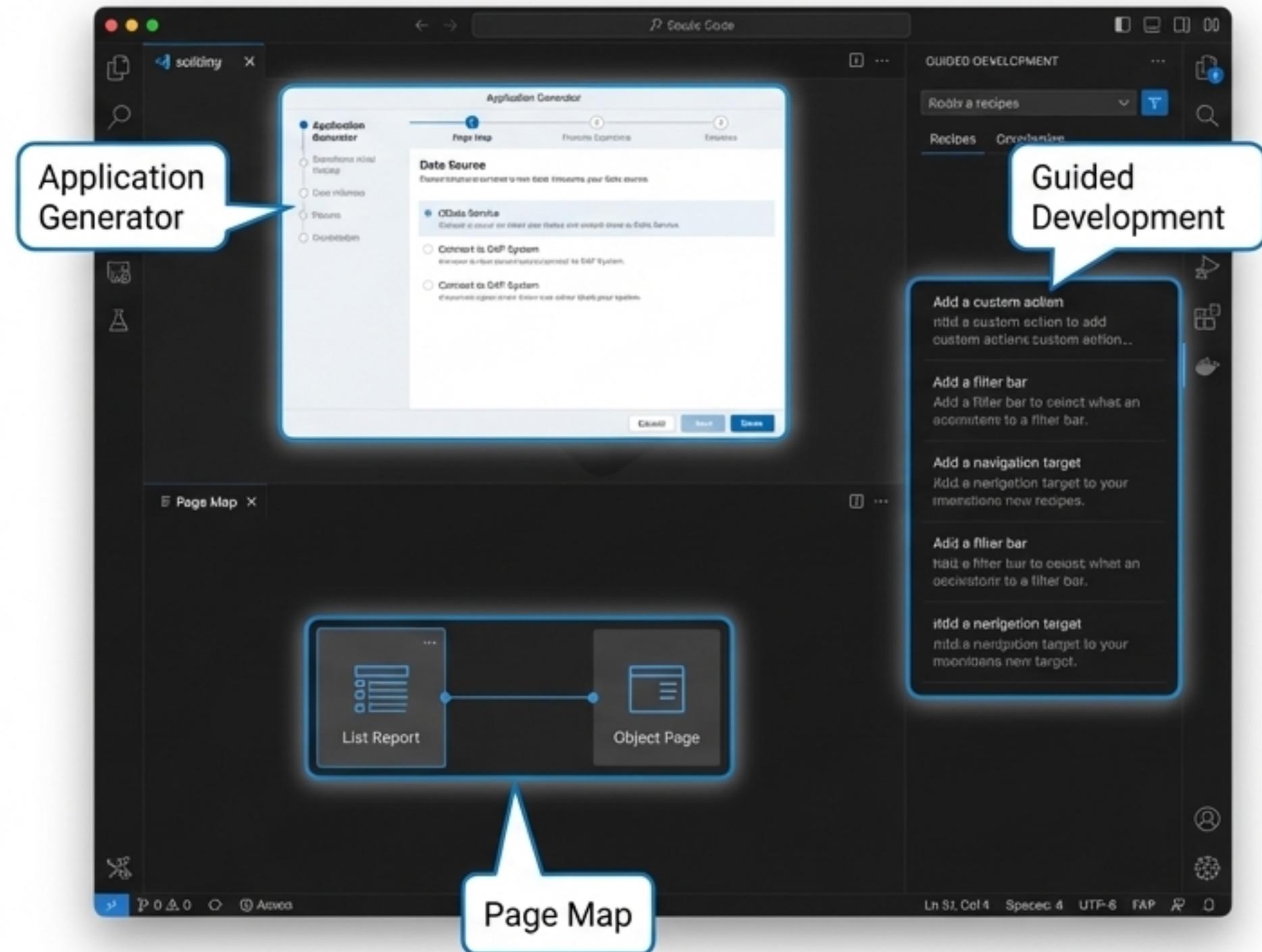
Acelerando o Desenvolvimento com Ferramentas Visuais

Ambientes Suportados

- SAP Business Application Studio (BAS):** Cloud IDE, pré-configurado.
- Visual Studio Code (VS Code):** Editor local com "SAP Fiori Tools - Extension Pack".

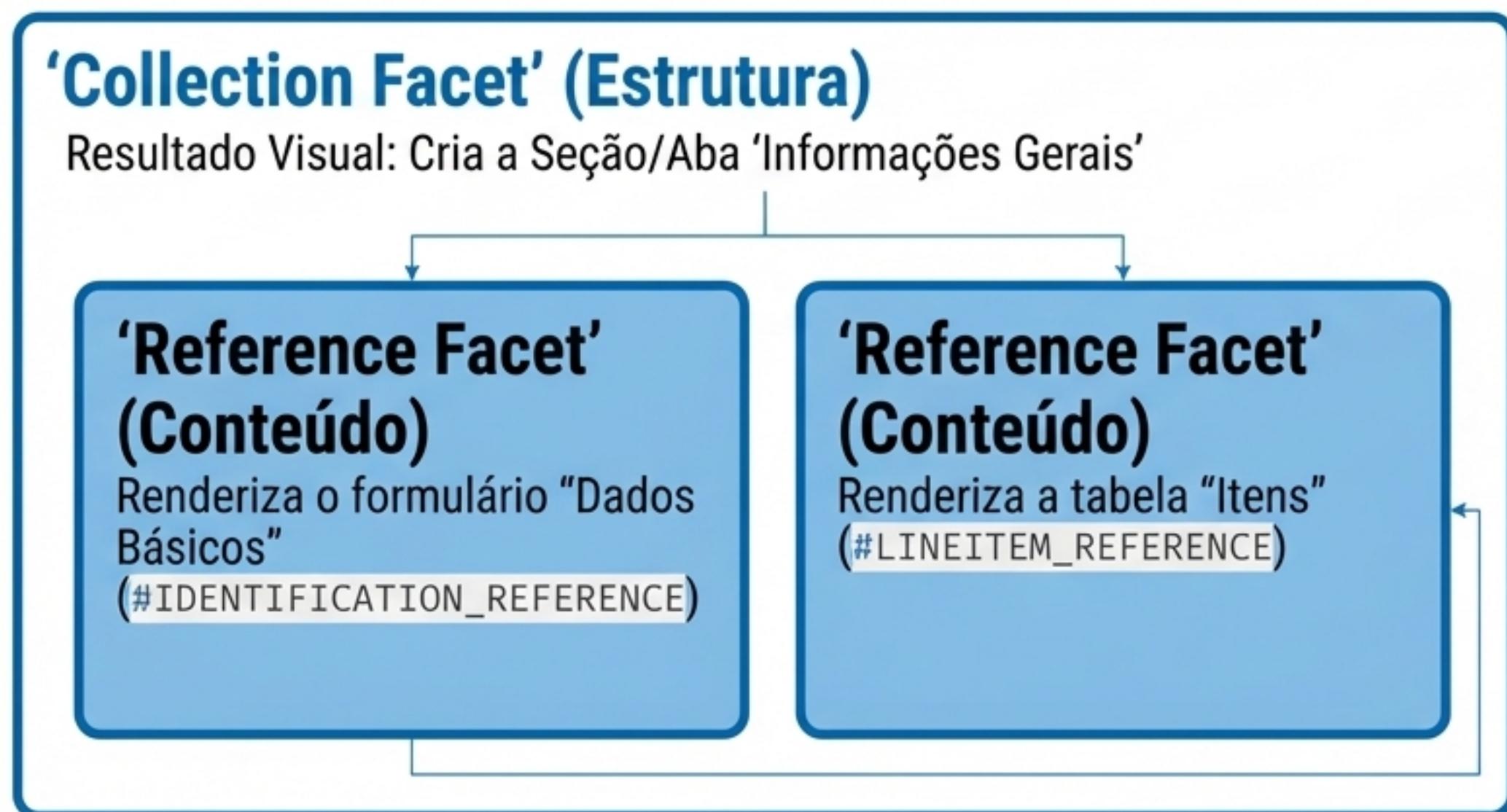
Ferramentas Essenciais

- Application Generator:** Cria a estrutura inicial do projeto via assistente.
- Page Map:** Visualiza a hierarquia de páginas e facilita extensões.
- Guided Development:** Biblioteca de "receitas" de código para tarefas comuns.



Facetas: Os Blocos de Construção da Object Page

Facetas (@UI.facet) são os elementos que dizem ao Fiori Elements como **organizar o conteúdo na Object Page**. Sem elas, a página fica vazia.



Descrição da Collection Facet

Propósito: Agrupar outras facetas.
Não renderiza conteúdo diretamente.



Descrição da Reference Facet

Propósito: Apontar para uma definição de conteúdo (formulários, tabelas) e renderizá-la.

Dando Vida à Interface: Feedback Visual e Ações



1. Criticality (Cores Semânticas)

Mapeia um valor numérico para cores que transmitem significado (Vermelho/Amarelo/Verde) para status, com ícones para acessibilidade (#WITH_ICON).



2. Actions (Ações de Negócio)

Adiciona botões na linha ou na barra de ferramentas (#FOR_ACTION) que disparam lógica de negócio definida no backend.



3. Importance (Responsividade)

Controla quais colunas se adaptam a telas menores, garantindo usabilidade em celulares (#HIGH, #MEDIUM, #LOW).

Visual de Comparação

Antes ►

Fiori				
Pedido	Data	Valor	Status	C ...
10030071	03/06/2022	7,00	Aprovado	
10000120	06/06/2022	20,00	Pendente	
10000133	12/03/2022	45,00	Rejeitado	
10003128	14/03/2023	0,00	Rejeitado	

◀ Depois

Fiori					
Pedido	Data	Valor	Status	Ações	C ...
10030071	03/06/2022	7,00	✓ Aprovado	Aprovar Rejeitar	
10000120	06/06/2022	20,00	✓ Aprovado	Aprovar Rejeitar	
10000133	12/03/2022	45,00	! Pendente	Aprovar Rejeitar	
10003137	14/03/2023	0,00	✗ Rejeitado	Aprovar Rejeitar	

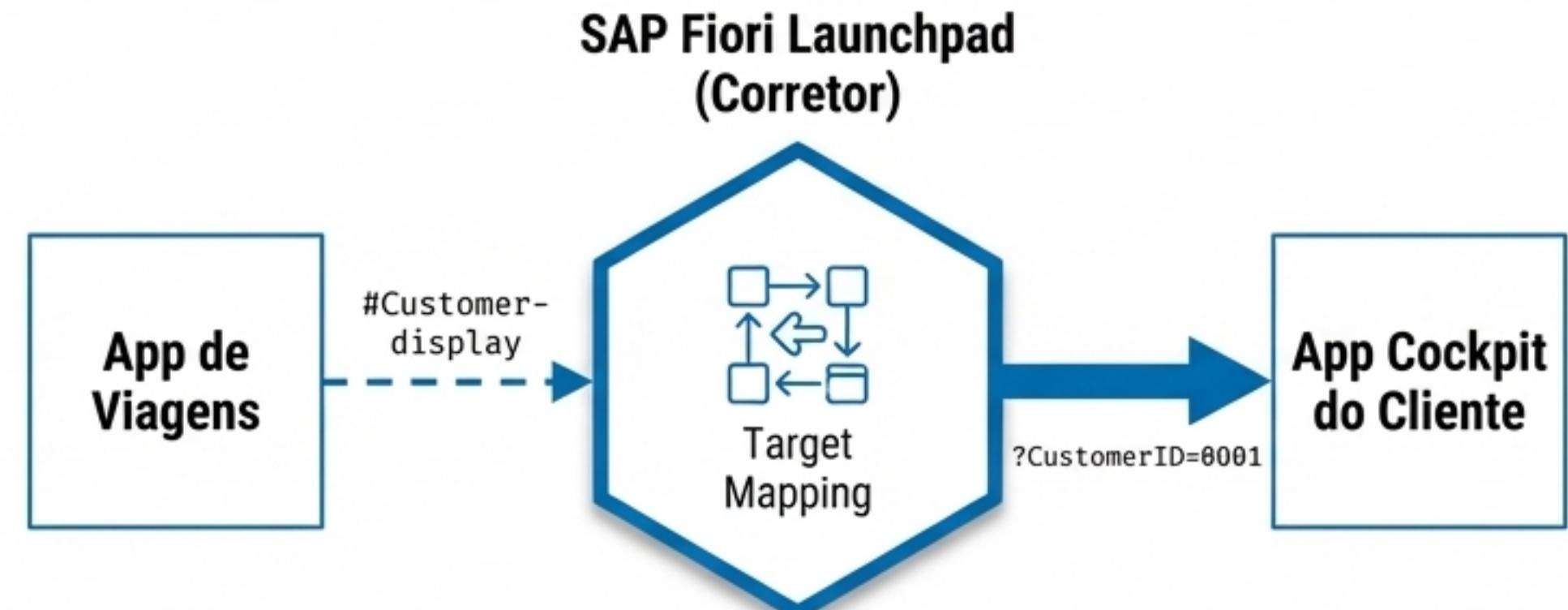
Conectando o Ecossistema com Navegação Inteligente

O Mecanismo: Navegação Baseada em Intenção

#SemanticObject-Action
(Ex: #Customer-display)

O app de origem não sabe qual app de destino será aberto. Ele apenas declara uma intenção ("Quero exibir um cliente"). O Fiori Launchpad atua como um "corretor" que resolve essa intenção com base na configuração e permissões do usuário (Target Mapping).

Usa-se @UI.lineItem: { type:
#FOR_INTENT_BASED_NAVIGATION } na coluna.

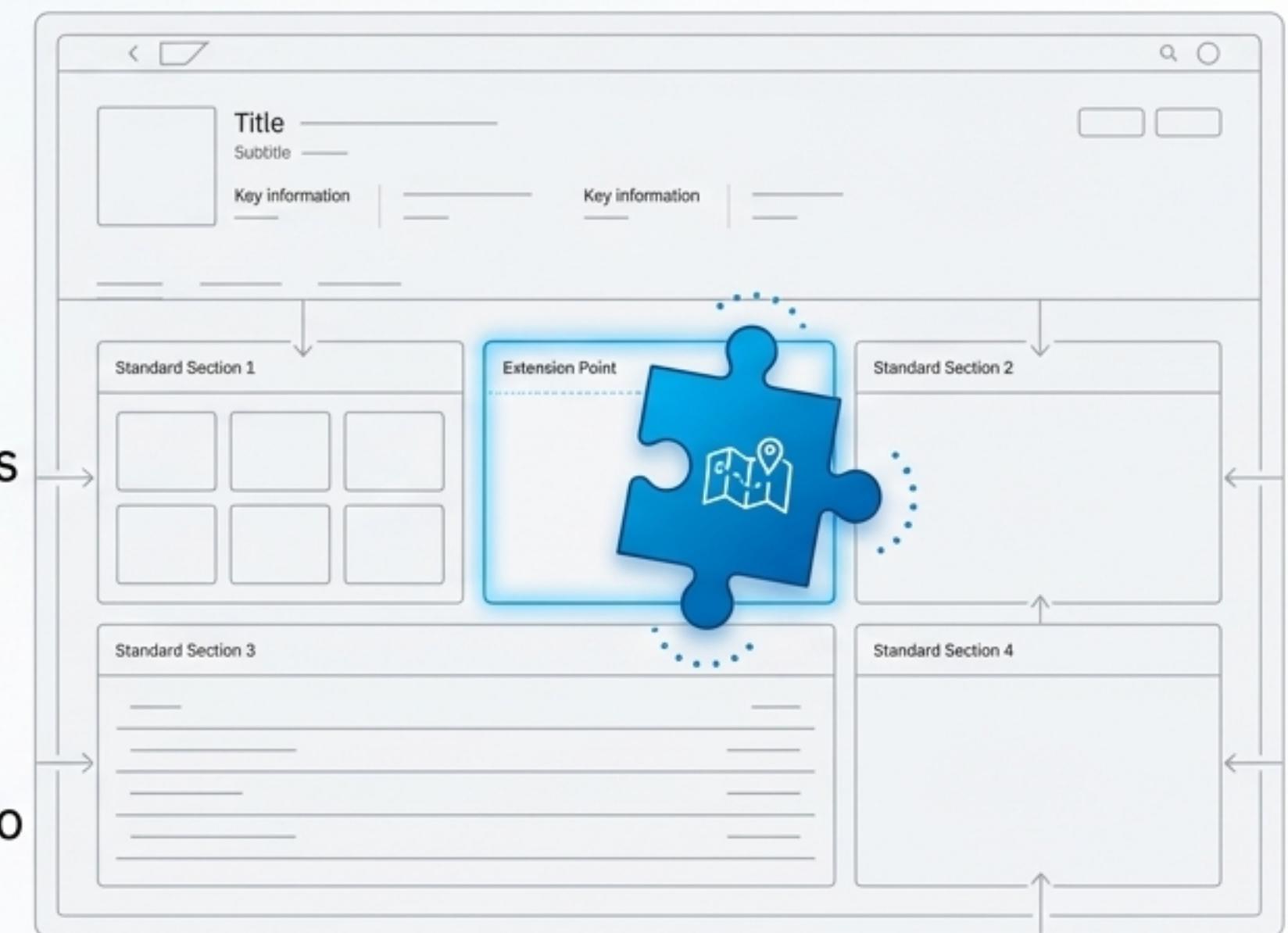


O Melhor dos Dois Mundos: Padrão com Flexibilidade

A Solução: Flexible Programming Model (FPM) permite manter 90% da aplicação padrão e “injetar” código customizado nos 10% que precisam de algo especial, de forma suportada e estável.

Ferramentas do FPM

- ➡ **Extension Points:** Locais seguros na estrutura padrão para inserir seções, colunas ou ações customizadas (Views e Fragments XML).
- ➡ **Building Blocks (Macros):** Componentes inteligentes (<macros:Table>) que trazem a funcionalidade 'metadata-driven' para dentro das suas seções customizadas.
- ➡ **Controller Extensions:** Arquivos JavaScript para adicionar lógica de frontend customizada, interagindo com a aplicação via ExtensionAPI.



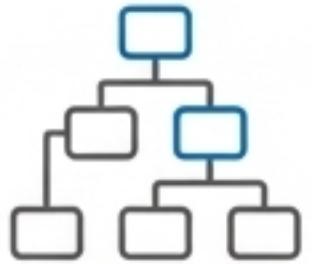
Da Sua Máquina ao Launchpad: O Processo de 'Go-Live'



Passo 1: Build & Deploy

As Fiori Tools executam um build (otimizando o código) e fazem o upload para o servidor ABAP.

Resultado: Código armazenado como uma **BSP Application**.



Passo 2: Configuração do Fiori Launchpad (FLP)

A configuração é feita em dois níveis para reutilização:

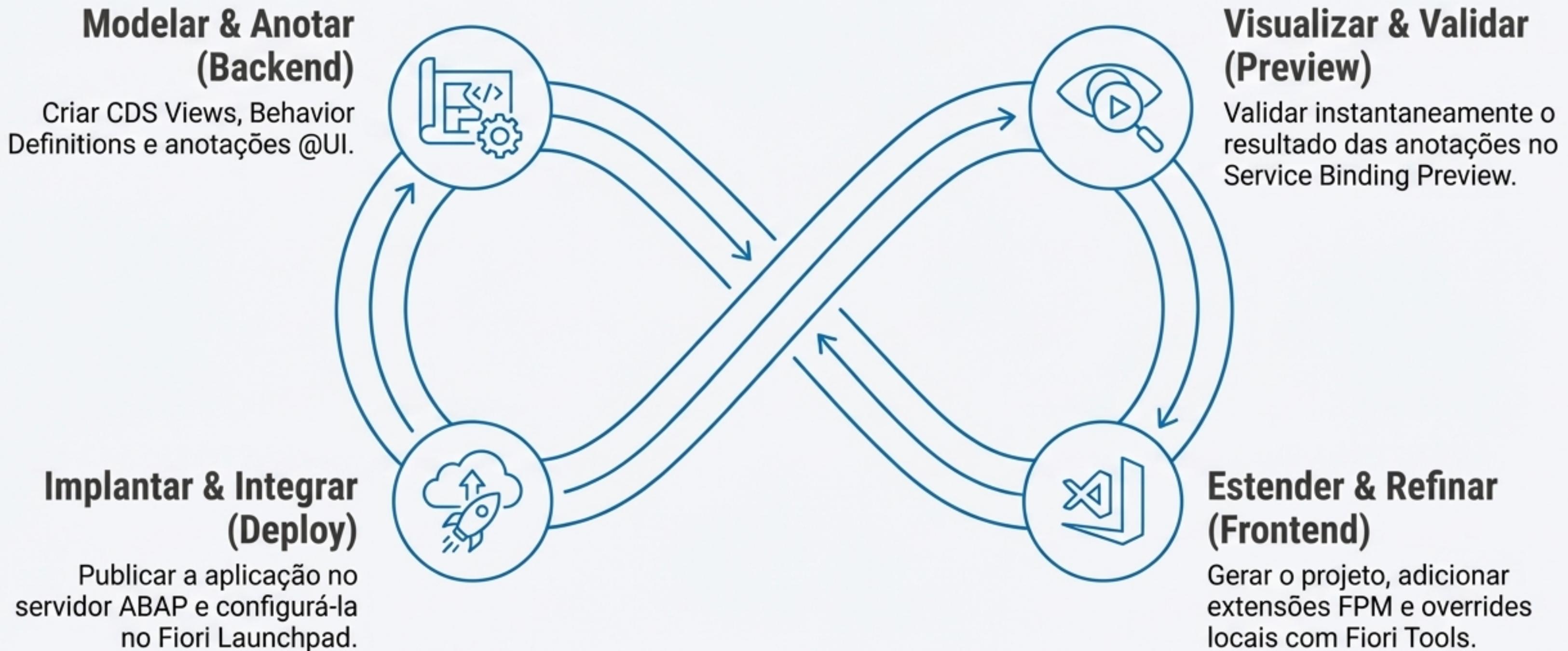
- **Technical Catalog:** Repositório com Tile + Target Mapping.
- **Business Catalog:** Agrupamento funcional que referencia o catálogo técnico.



Passo 3: Atribuição de Acesso (IAM)

O **Business Catalog** é atribuído a uma **Business Role**. A Role é atribuída ao **Usuário Final**.

O Workflow do Desenvolvedor Moderno

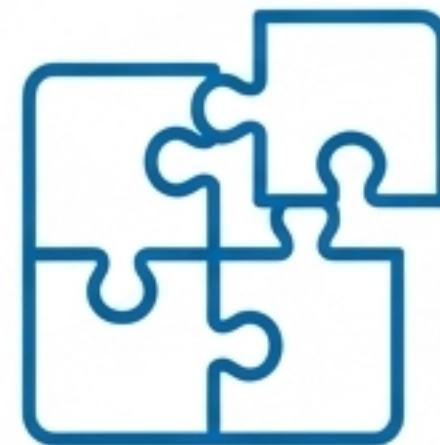


O Que Lembrar Desta Jornada Jornada



O Backend é o Rei

A verdade da UI reside nas anotações CDS. A maior parte da definição da interface deve acontecer no backend para garantir consistência e reutilização.



Padrão Primeiro, Flexibilidade Sob Demanda

Abrace a eficiência do Fiori Elements para 90% dos casos. Use o Flexible Programming Model (FPM) como uma 'válvula de escape' cirúrgica para requisitos complexos.



Ferramentas Aceleram, Não Substituem

Use o poder visual das SAP Fiori Tools para eliminar trabalho manual, mas entenda a arquitetura subjacente que elas manipulam.



Desacoplamento é Poder

A Navegação Baseada em Intenção e a hierarquia de Catálogos garantem a manutenibilidade do ecossistema. Seus aplicativos não quebram quando outros são atualizados.

A Arquitetura Completa: Da Lógica de Negócio ao Acesso do Usuário

