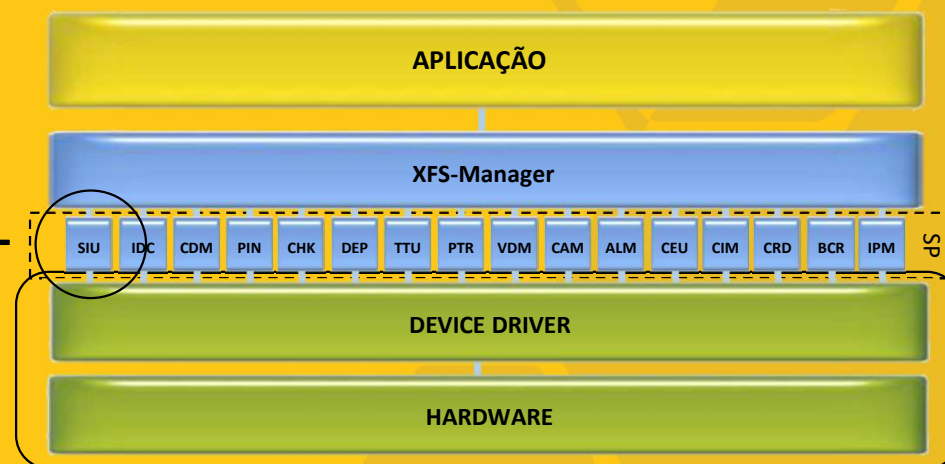


# XFS

## Visão geral

**Nosso alvo:** cada dispositivo bancário precisa de um Service Provider (SP) para que a aplicação cliente consiga controlá-lo.

Os SPs devem ser construídos em conformidade com uma versão do XFS (ex.: 3.10). Ou seja, funções, parâmetros e retornos devem estar adequados a especificação contida na versão XFS escolhida.



### Diversos fabricantes (vendors):

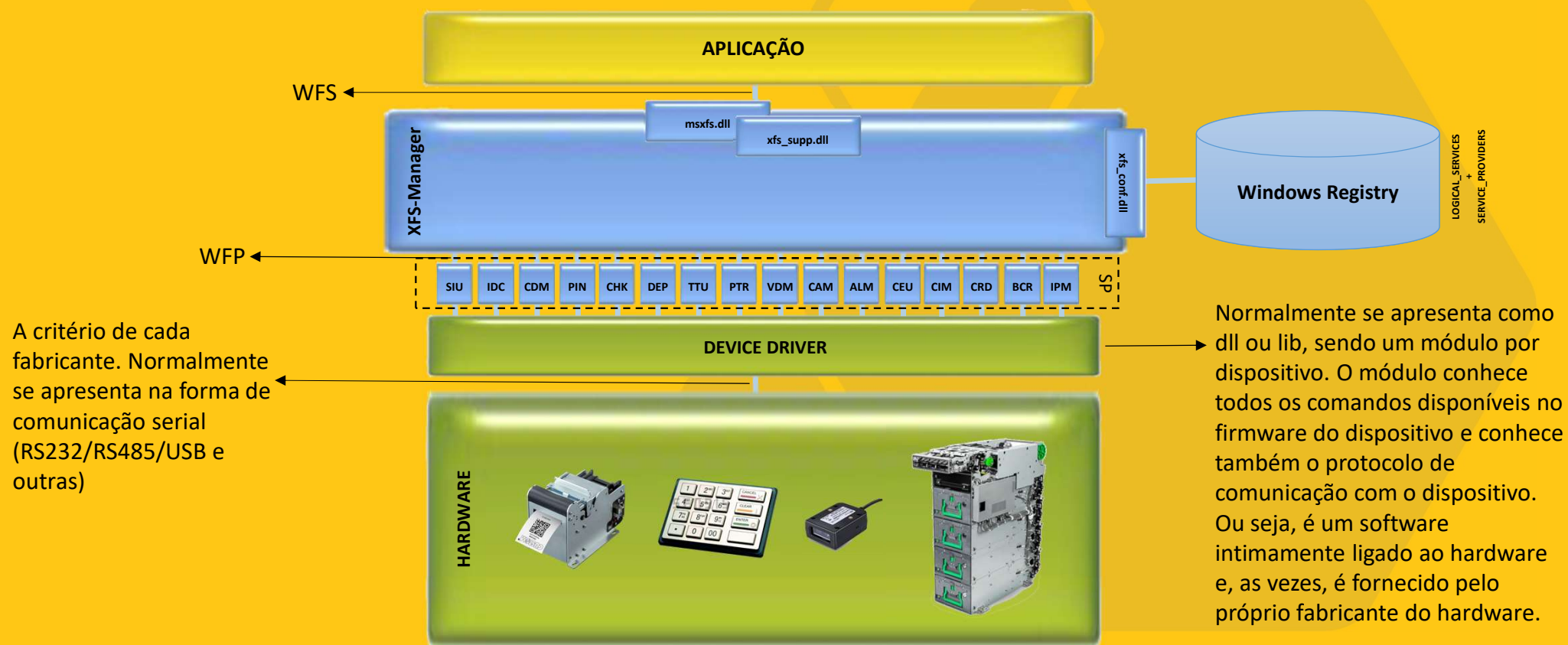
- Diebold;
- NCR;
- Hyosung;
- Fujitsu;
- Perto;
- ...

**Dispositivos bancários:** dispensador de cédulas, depositário de envelopes, placa de leds e sensores, leitora de cartões, leitor de código de barras, painel do operador, impressora de recibos, teclado numérico, e etc. Cada dispositivo precisa de um device driver que conhece a especificidade do dispositivo:



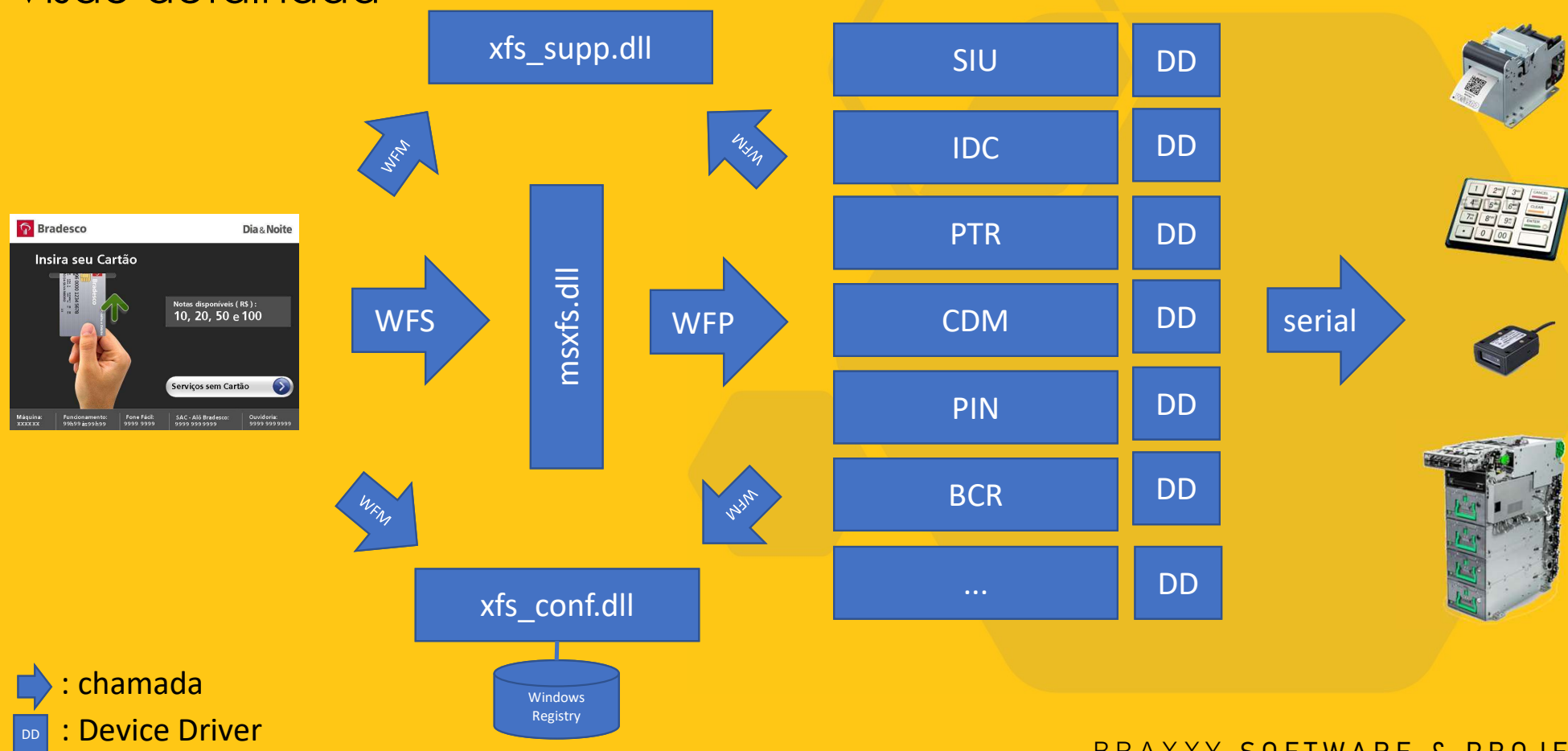
# XFS

## Visão detalhada



# XFS

## Visão detalhada



# XFS

## Prática: planejando o desenvolvimento do SP

### O que incluir no projeto do Service Provider?

- **Headers:**

- **XFSSPI.H:** declaração das funções WFP;
- **XFSXXX.H:** substituir XXX pela classe que se deseja implementar no SP. Esse header possui definições de estrutura de dados, códigos de retorno e demais informações básicas específicas da classe XFS. Ou seja, não há definição de funções aqui!;
- **XFSAADMIN.H:** declaração das funções WFM de suporte; (adicionado via XFSSPI.H)
- **XFSCONF.H:** declaração das funções WFM de acesso a configuração XFS; (adicionado via XFSSPI.H)
- **XFSAPI.H:** definições de constantes e estruturas de dados comuns a todos os dispositivos; (adicionado via XFSSPI.H)  
(VC++ Directories -> Include Directories)

- **Lib:**

- **xf\_conf.lib:** implementação das funções WFM de acesso a configuração XFS (Windows Registry);
- **xf\_supp.lib:** implementação das funções WFM de suporte;  
(VC++ Directories -> Library Directories / Linker -> Input -> Additional Dependencies)

- E cadê a implementação das funções WFP declaradas no XFSSPI.H?

- Essa é a implementação que você, programador, deve fazer em sua dll!!!
- Crie um arquivo DEF file e adicione o nome de sua DLL (na sessão LIBRARY) e as funções WFP implementadas pela DLL (na sessão EXPORTS);

(Linker -> Input -> Module Definition File)

Abra os headers citados aqui:

- XFS\SDK\INCLUDE\XFSAADMIN.H
- XFS\SDK\INCLUDE\XFSCONF.H
- XFS\SDK\INCLUDE\XFSSPI.H

Abra também os headers das definições de classes

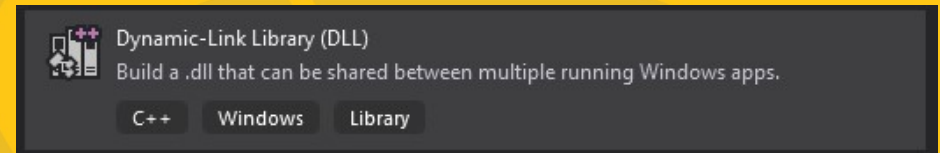


# XFS

## Prática: desenvolvendo o primeiro SP

### Criar projeto e solução no Visual Studio 2019:

- **Tipo:** Dynamic-Link Library (DLL);
- **Nome do projeto:** spsiu
- **Nome da solução:** xfs
- **Classe XFS:** SIU
- **Detalhes:**
  1. Ajustar includes e imports conforme slide anterior;
  2. Criar o arquivo DEF conforme slide anterior;
  3. Incluir as funções WFP;
  4. Retornar WFS\_ERR\_INTERNAL\_ERROR em todas as funções;
  5. Incluir nas funções uma chamada ao WFMOuputTraceData. Escreva qualquer dado para fins de teste;
  6. Fazer o build da DLL;
  7. Confirmar as funções exportadas com o DLL Explorer Viewer ou DUMPBIN;
  8. Configurar um serviço XFS no Regedit para encontrar sua nova DLL; Configurar caminho do arquivo trace;



# XFS

## Prática: planejando o desenvolvimento da APP

### O que incluir no projeto da APP cliente?

- **Headers:**
  - **XFSADMIN.H:** declaração das funções WFM de suporte;
  - **XFSCONF.H:** declaração das funções WFM de acesso a configuração XFS;
  - **XFSAPI.H:** declaração das funções WFS de acesso aos dispositivos;
- **Lib:**
  - **xfp\_conf.lib:** implementação das funções WFM de acesso a configuração XFS (Windows Registry);
  - **xfp\_supp.lib:** implementação das funções WFM de suporte;
  - **msxfp.lib:** implementação das funções WFS de acesso aos dispositivos;

Abra os headers citados aqui:

- XFS\SDK\INCLUDE\XFSAPI.H



# XFS

## Prática: desenvolvendo o primeiro APP XFS client

### Criar projeto no Visual Studio 2019:

- **Tipo:** Console App
- **Nome do projeto:** xfstest
- **Nome da solução:** xfs
- **Detalhes:**
  1. Ajustar includes e imports conforme slide anterior;
  2. Fazer o include do windows.h
  3. Fazer chamada ao WFSStartup;
  4. Fazer chamada ao WFSOpen, chamando o seu serviço XFS criado anteriormente;
  5. Fazer chamada ao WFSCleanup;
  6. Usar uma ferramenta de testes XFS para tentar chamar o seu novo serviço XFS.

