

A Evolução dos Sistemas Operacionais

Do Mainframe à Inteligência Ambiente: Uma Jornada Arquitetural

EXHIBIT 01: CARTÃO PERFURADO (1950)

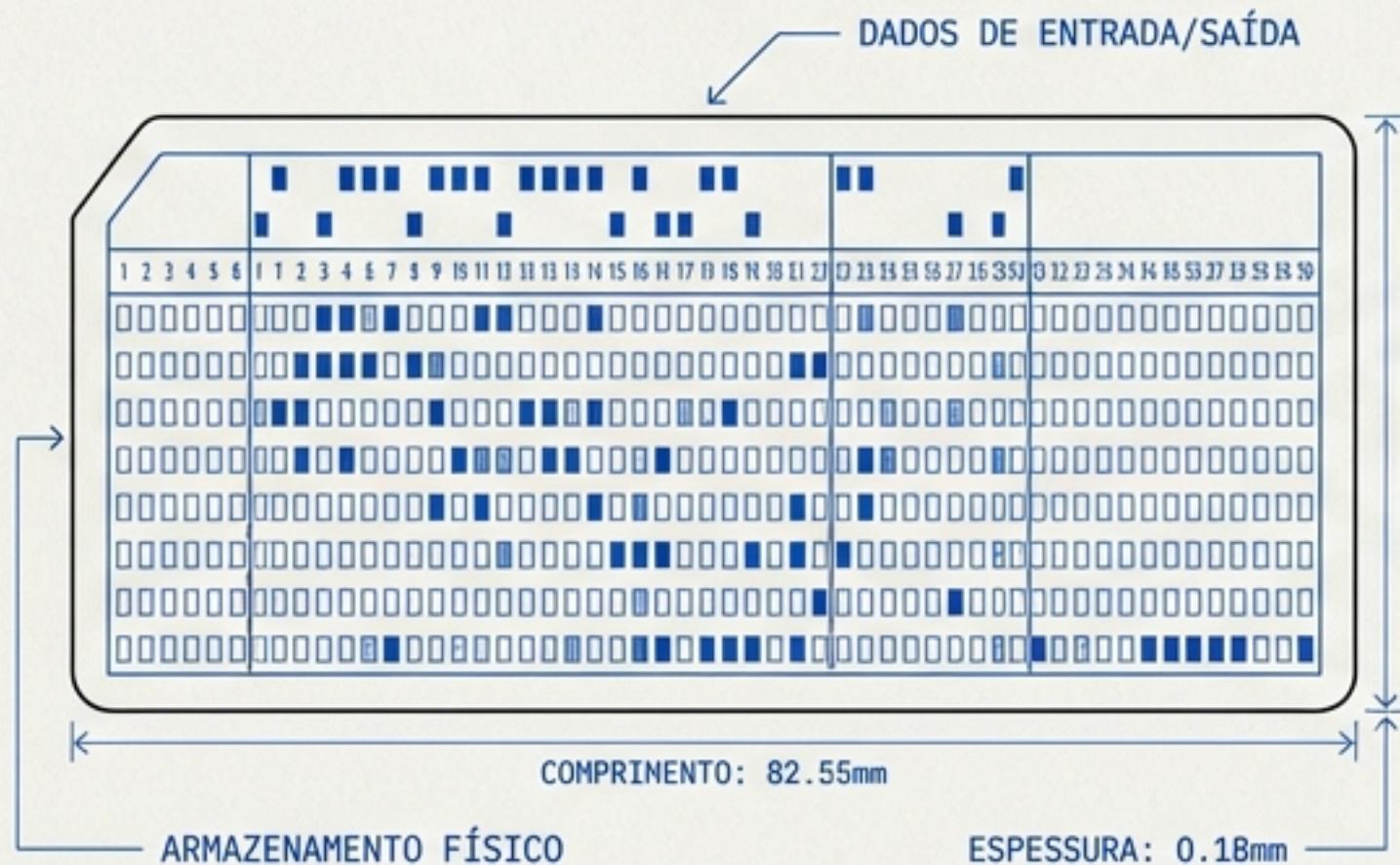
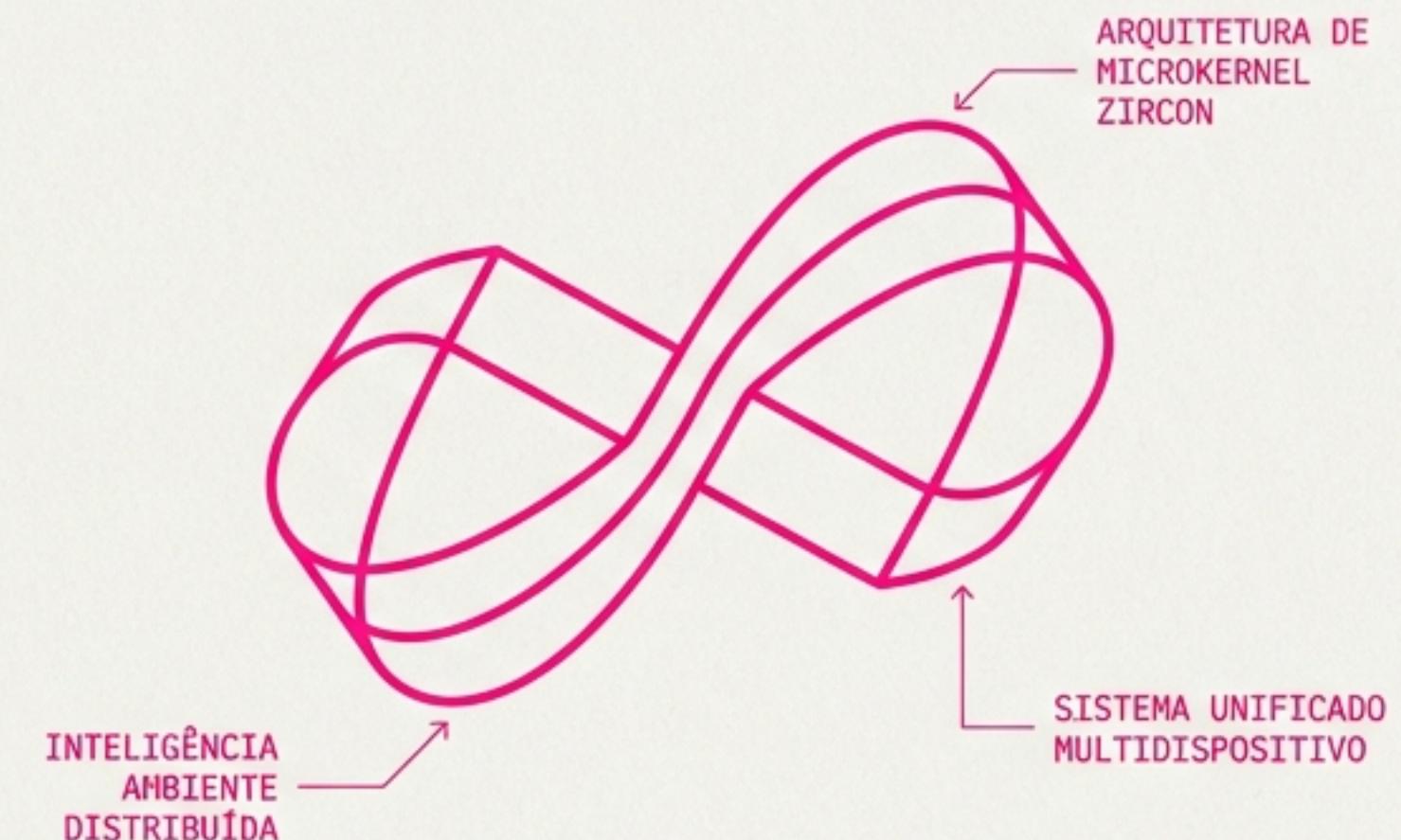
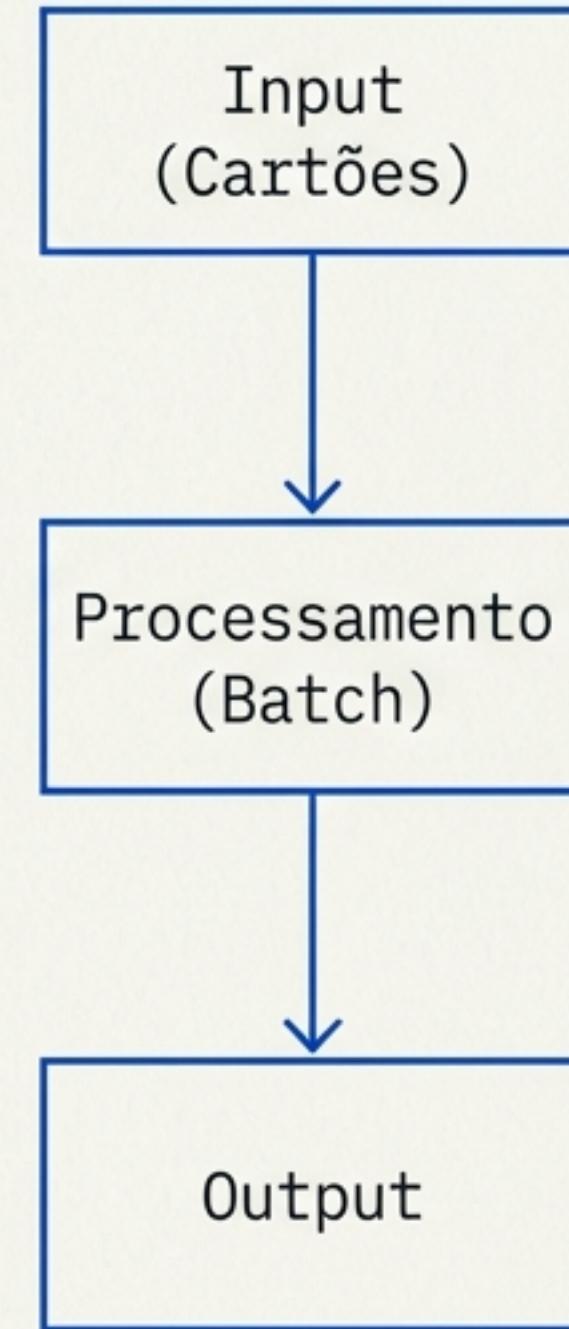
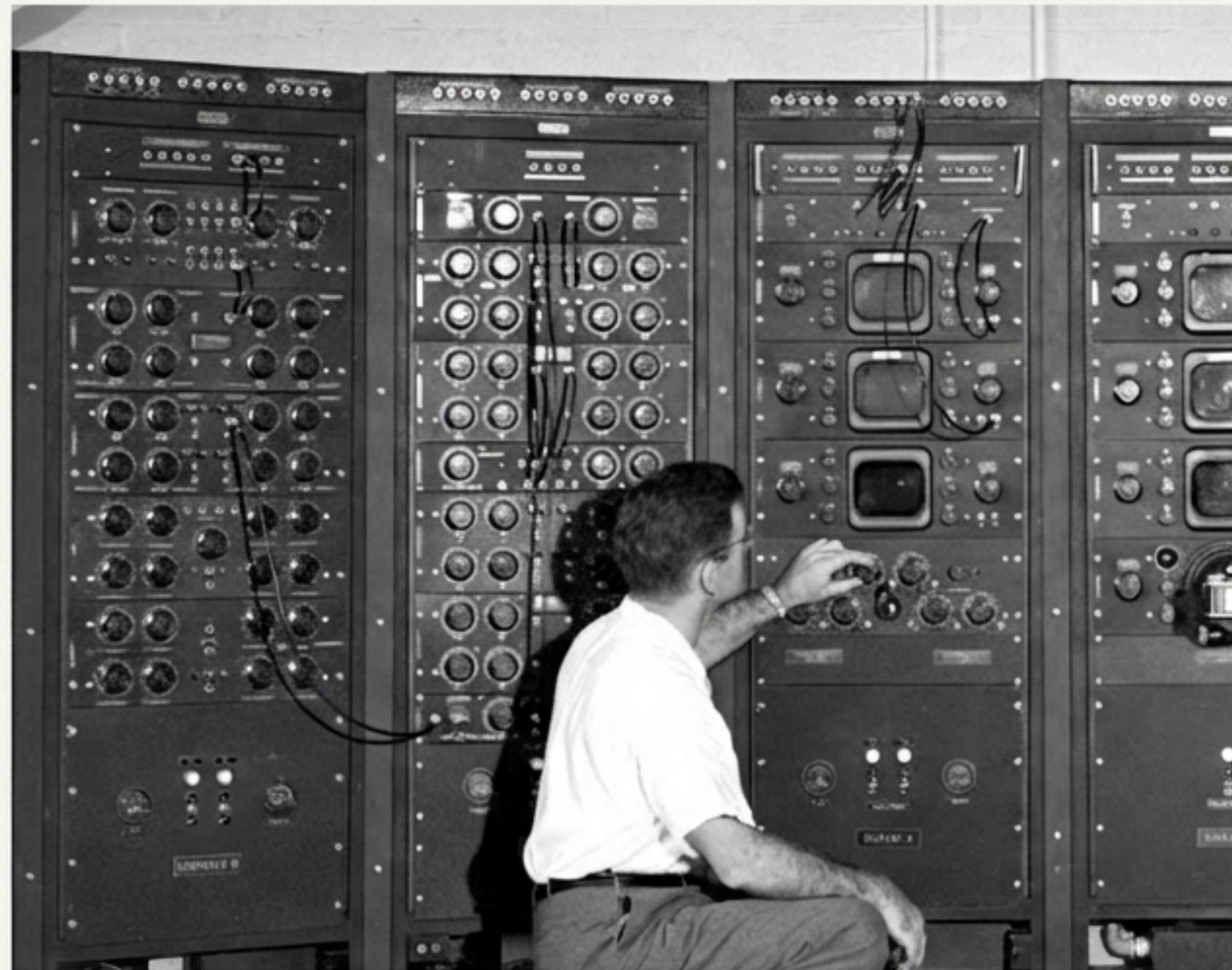


EXHIBIT 02: FUCHSIA OS (FUTURO)



A Era do Hardware Nu (1950s - 1960s)

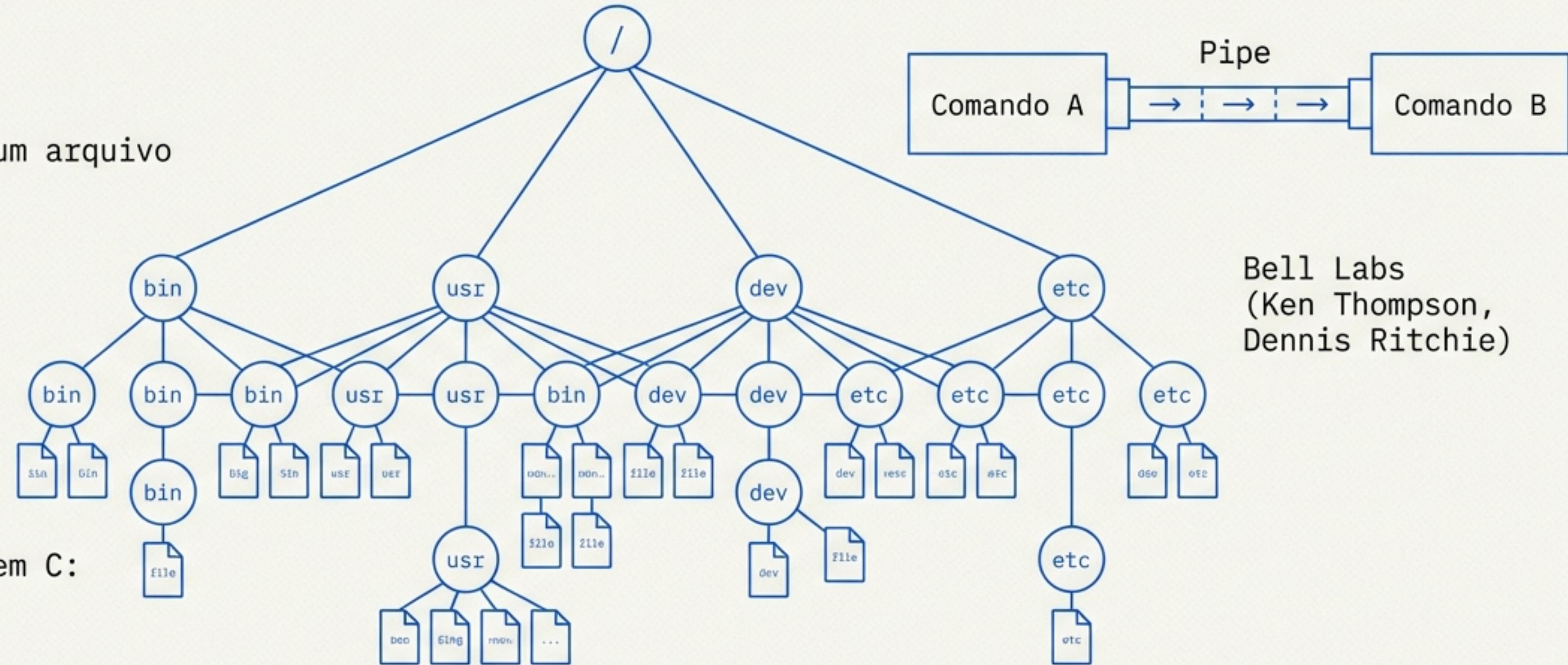


“No início, o programador era o sistema operacional.”

- Computadores Mainframe: GM-NAA I/O (1956)
- Hardware escasso, tempo humano barato
- Processamento sequencial (Batch)

O Nascimento do Padrão: UNIX e Time-Sharing (1970s)

Filosofia: Tudo é um arquivo



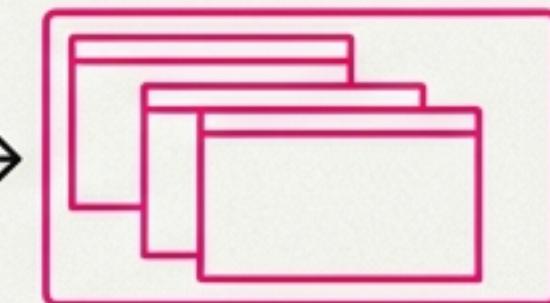
Criação da Linguagem C:
Portabilidade

A abstração começa: O SO se descola do hardware.

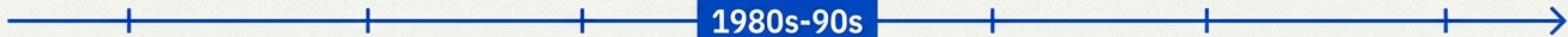
A Revolução da Interface: O Computador Pessoal



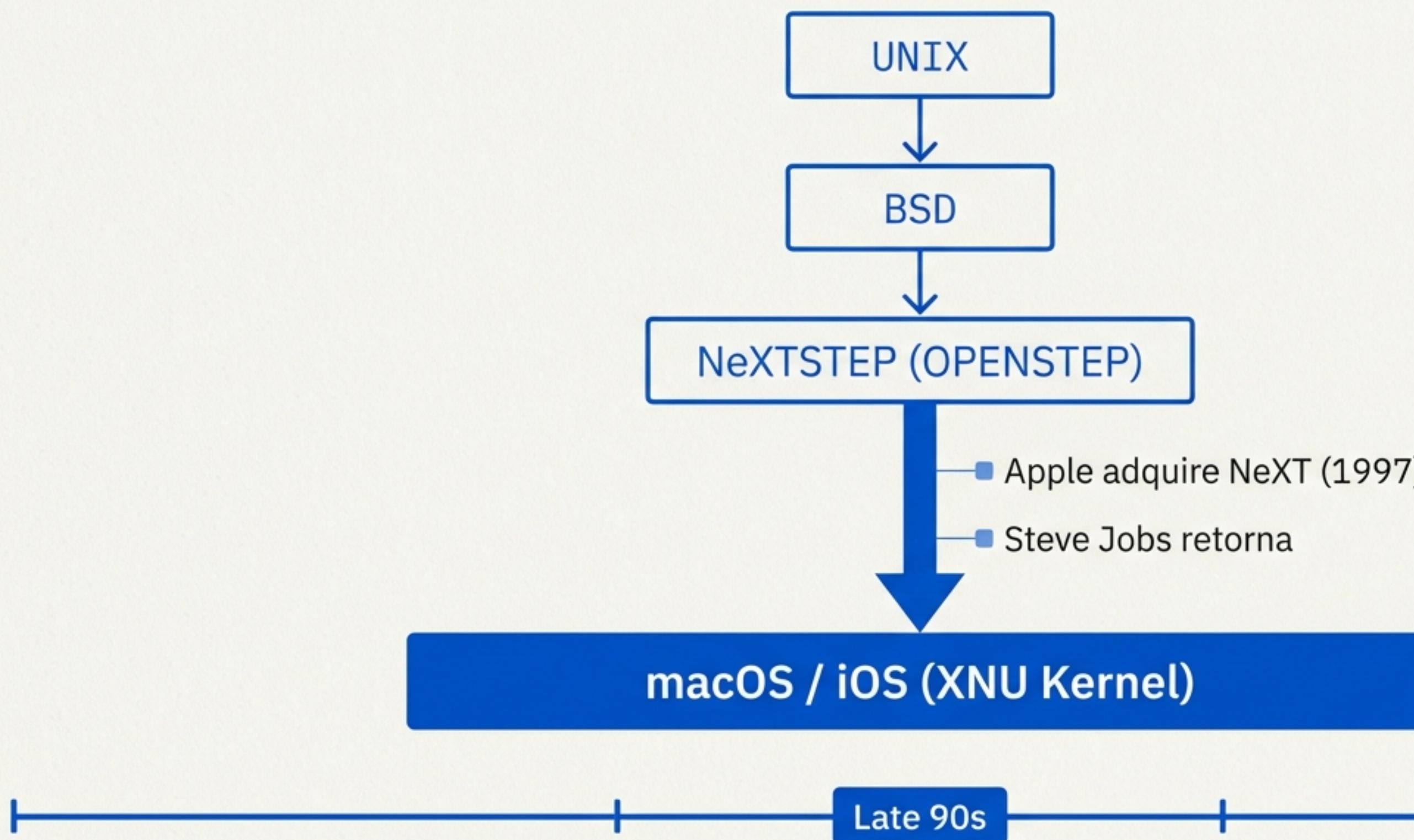
> C:_



- 1980s-90s: Apple Macintosh & Windows
- De Linha de Comando para GUI
- Foco: Eficiência do Usuário > Eficiência da Máquina

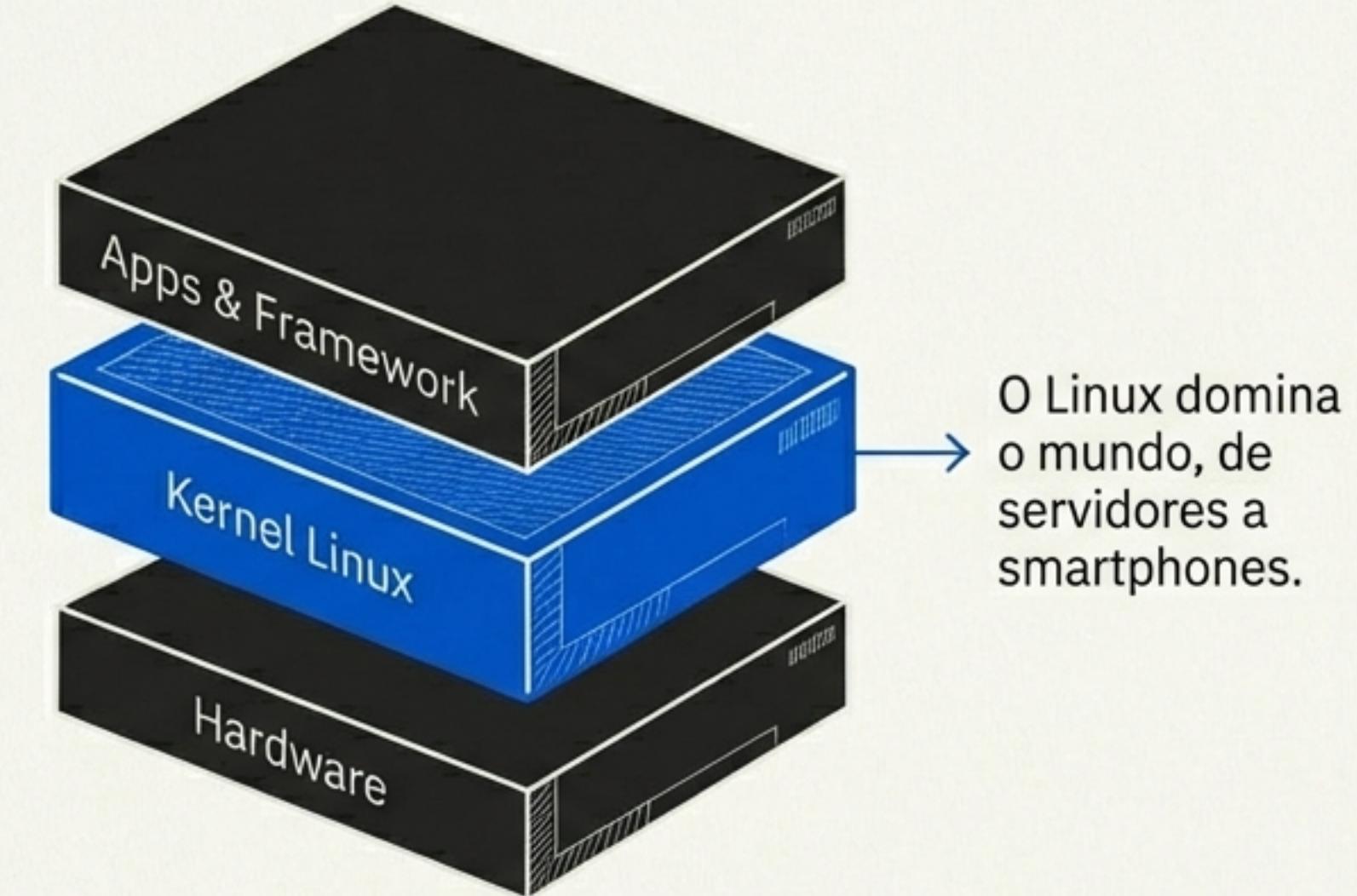


O Legado NeXTSTEP: A Gênese do macOS Moderno



O macOS moderno é um UNIX certificado, escondido sob uma interface polida.

A Era da Mobilidade: Android e iOS

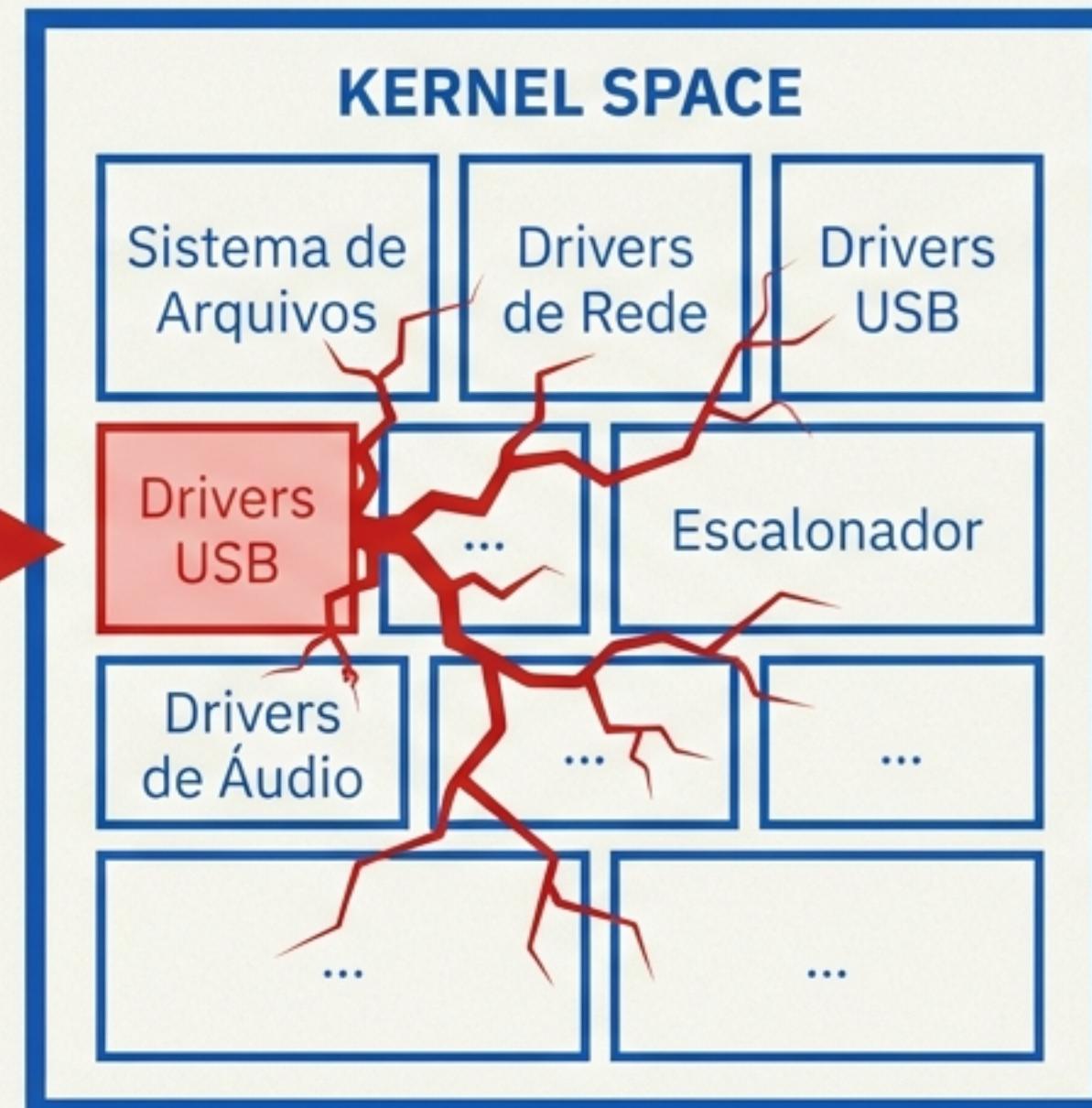


- Android: Baseado em Kernel Linux
- iOS: Baseado em Darwin/XNU
- Desafio: Código legado de décadas em dispositivos de bolso.

2007-Present

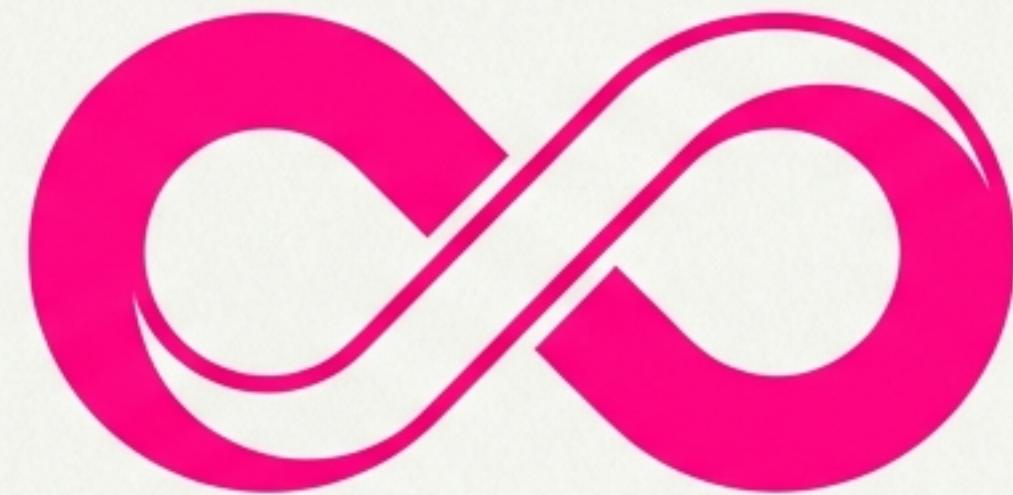
O Problema dos Kernels Monolíticos

**Falha Única =
Falha Total
(Kernel Panic)**



- Tudo roda com privilégio máximo
- Milhões de linhas de código
- Difícil auditoria de segurança

Introduzindo o Fuchsia



NÃO É LINUX.



Seguro



Atualizável



Inclusivo

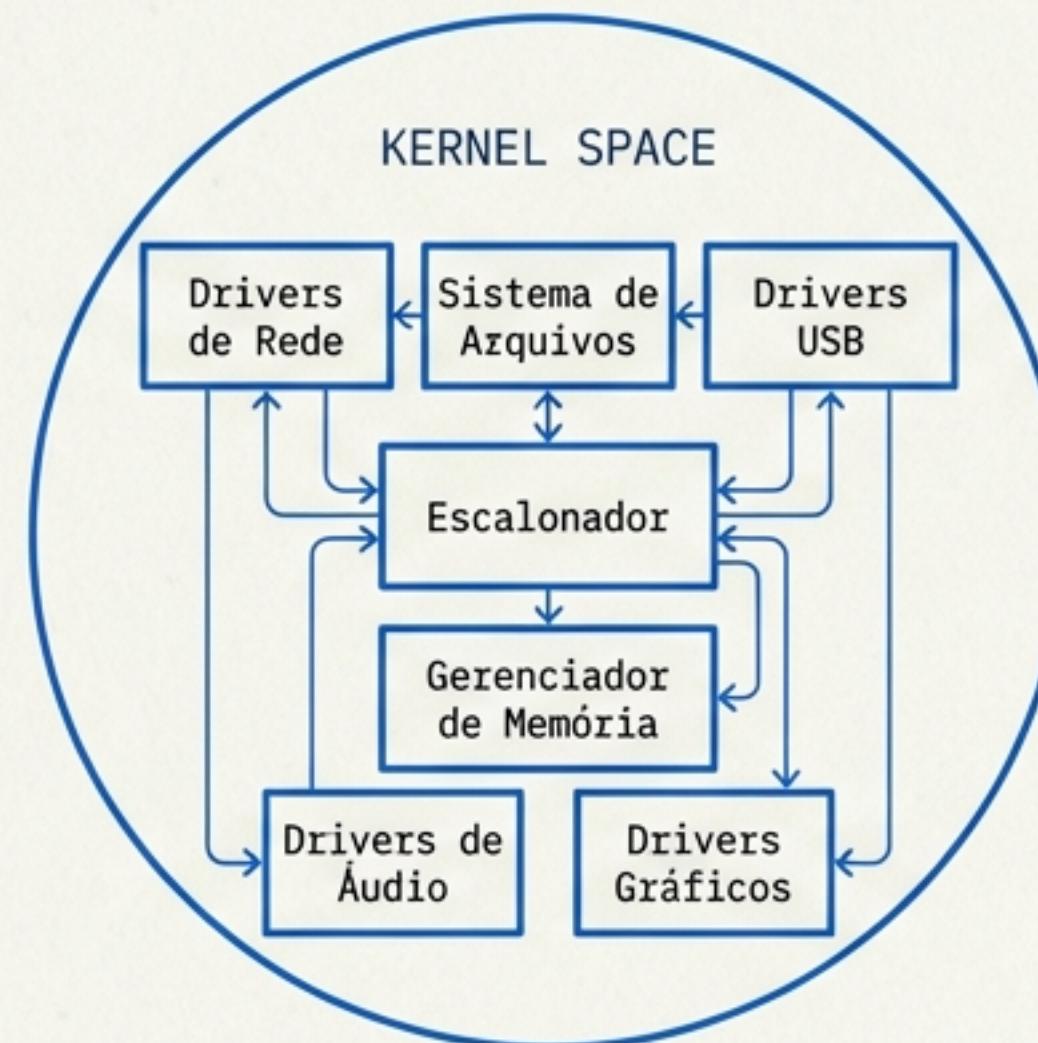


Performático

Um recomeço arquitetural (clean sheet) do Google.
Open-source desde o dia 1. Lançado no Nest Hub (2021).

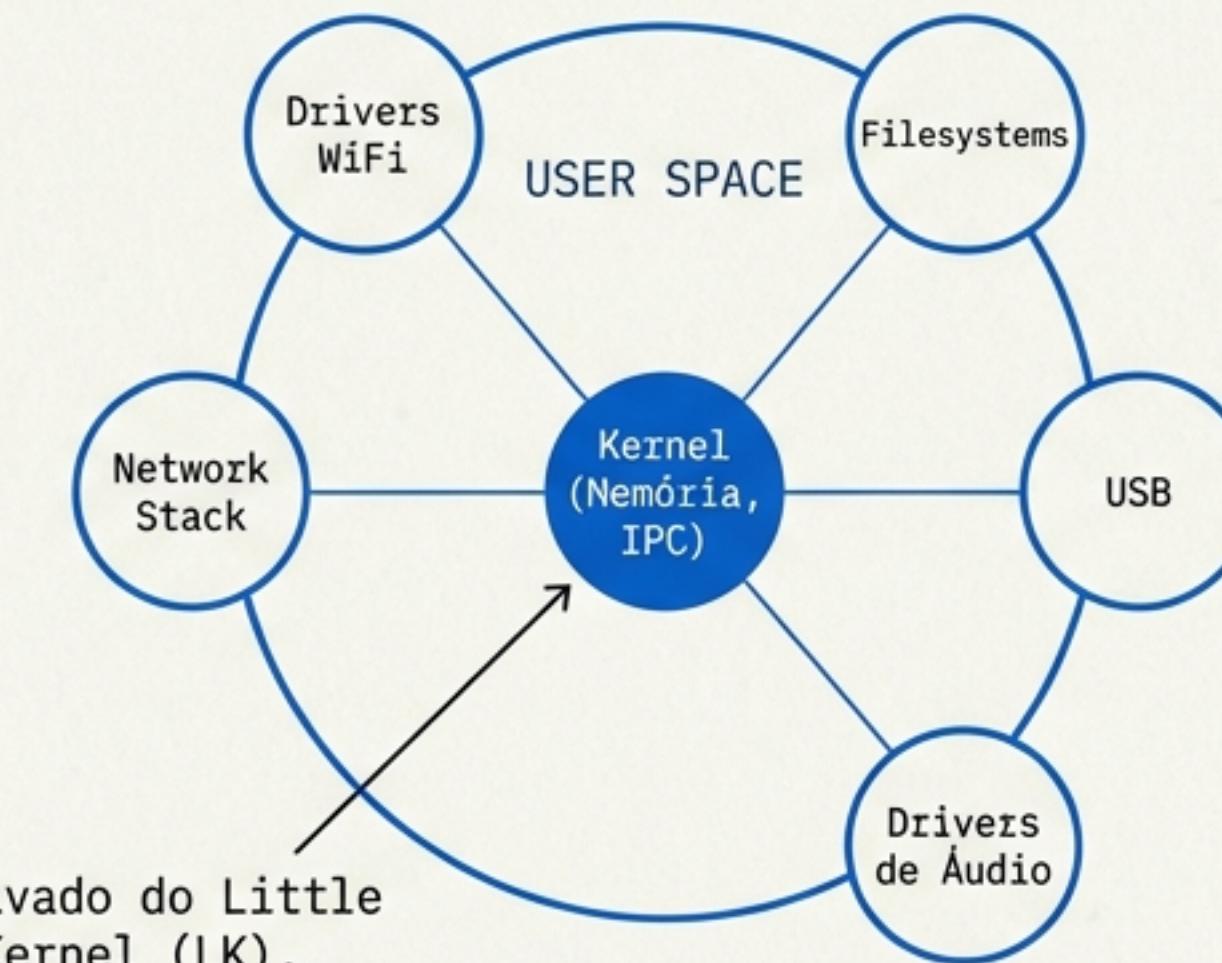
Sob o Capô: O Microkernel Zircon

Monolítico (Linux)



Tudo em um espaço de memória único.

Microkernel (Zircon)

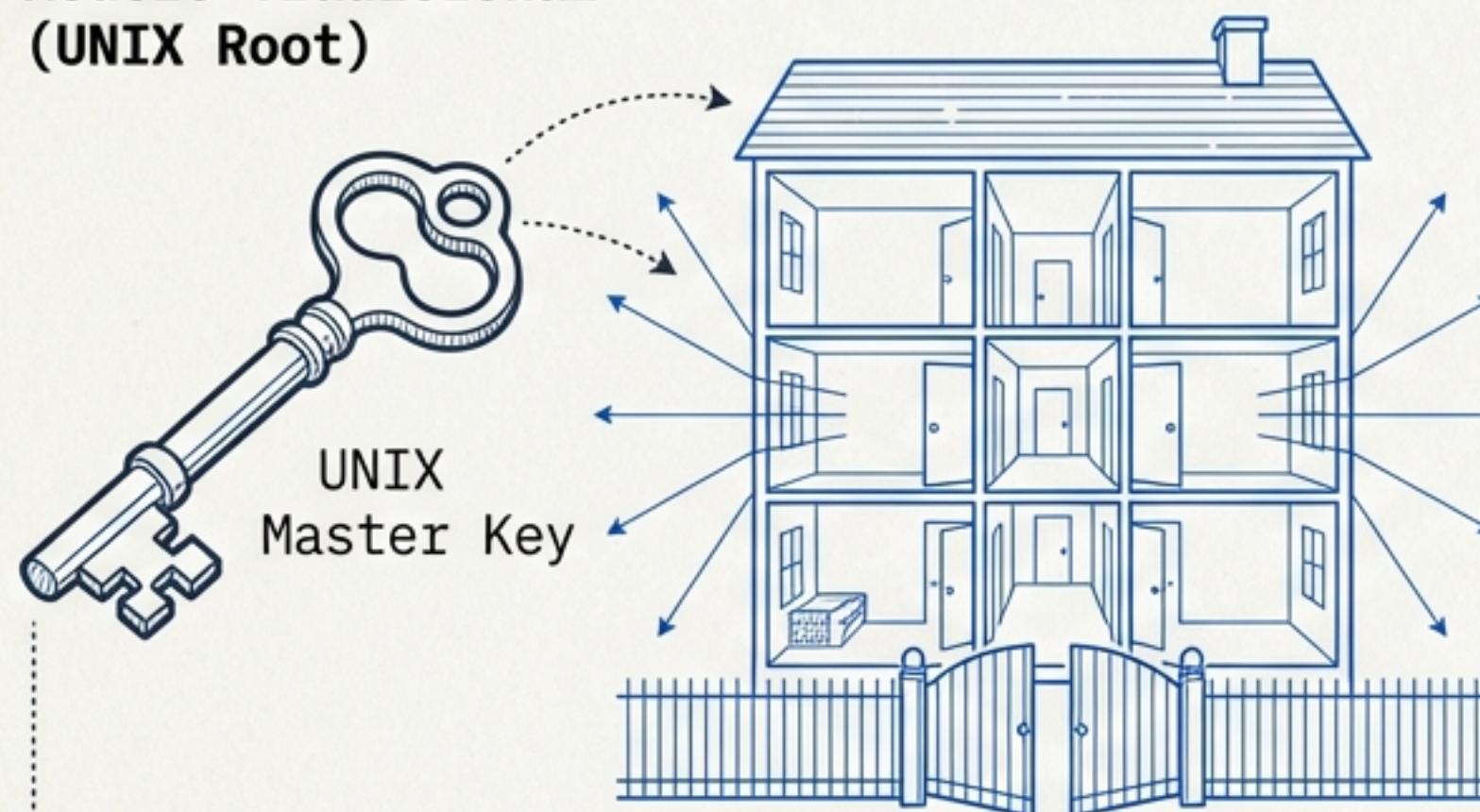


Derivado do Little Kernel (LK).
Escrito em C++.

Insight: Se um driver trava, o kernel reinicia apenas o driver, não o sistema.

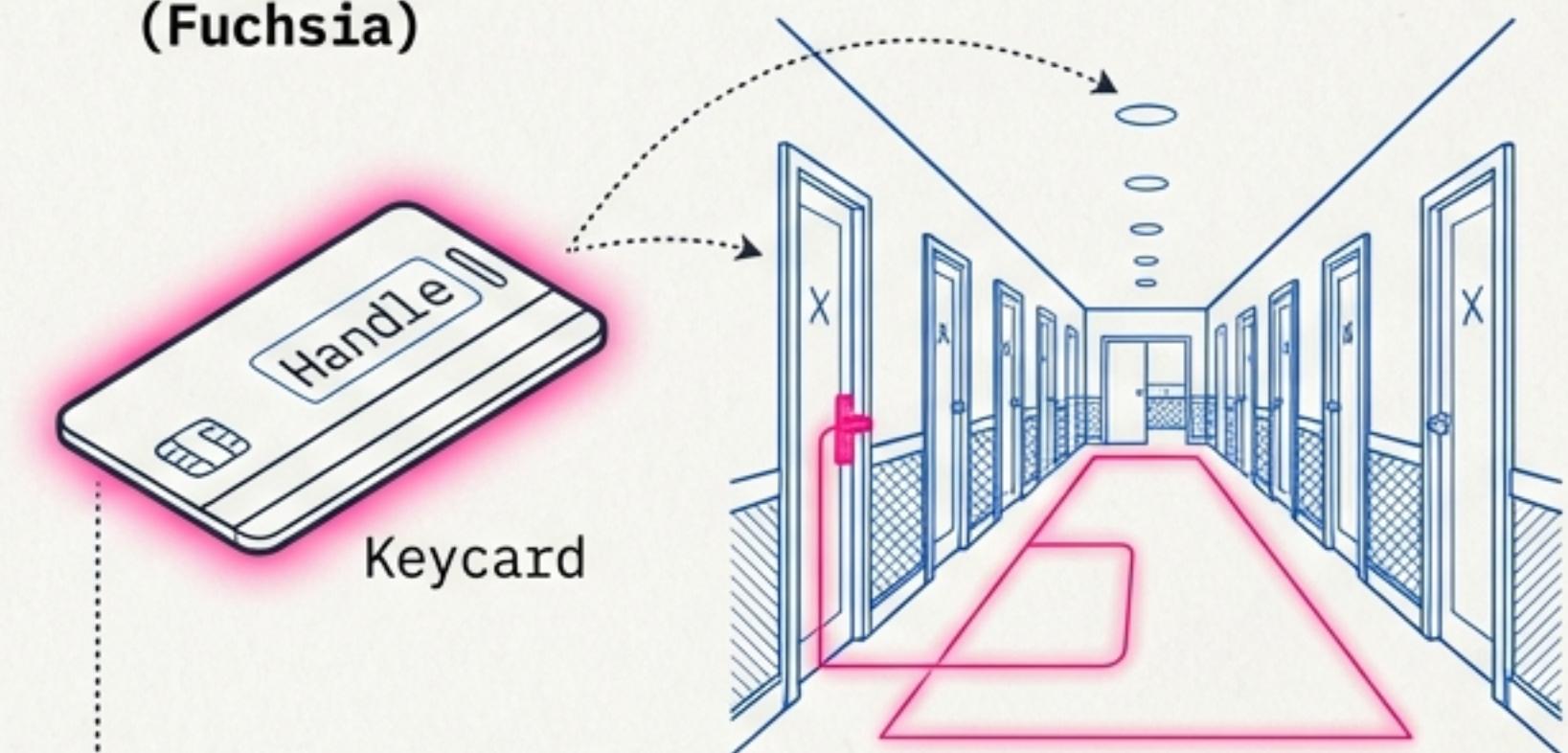
Segurança Baseada em Capacidades (Capability-Based)

**Modelo Tradicional
(UNIX Root)**



- Acesso irrestrito. Uma única falha compromete todo o sistema.

**Modelo Capability
(Fuchsia)**



- Princípio do Menor Privilégio: Nenhum componente tem acesso a nada, a menos que receba uma “chave” específica.
- Não existe usuário Root onipotente.

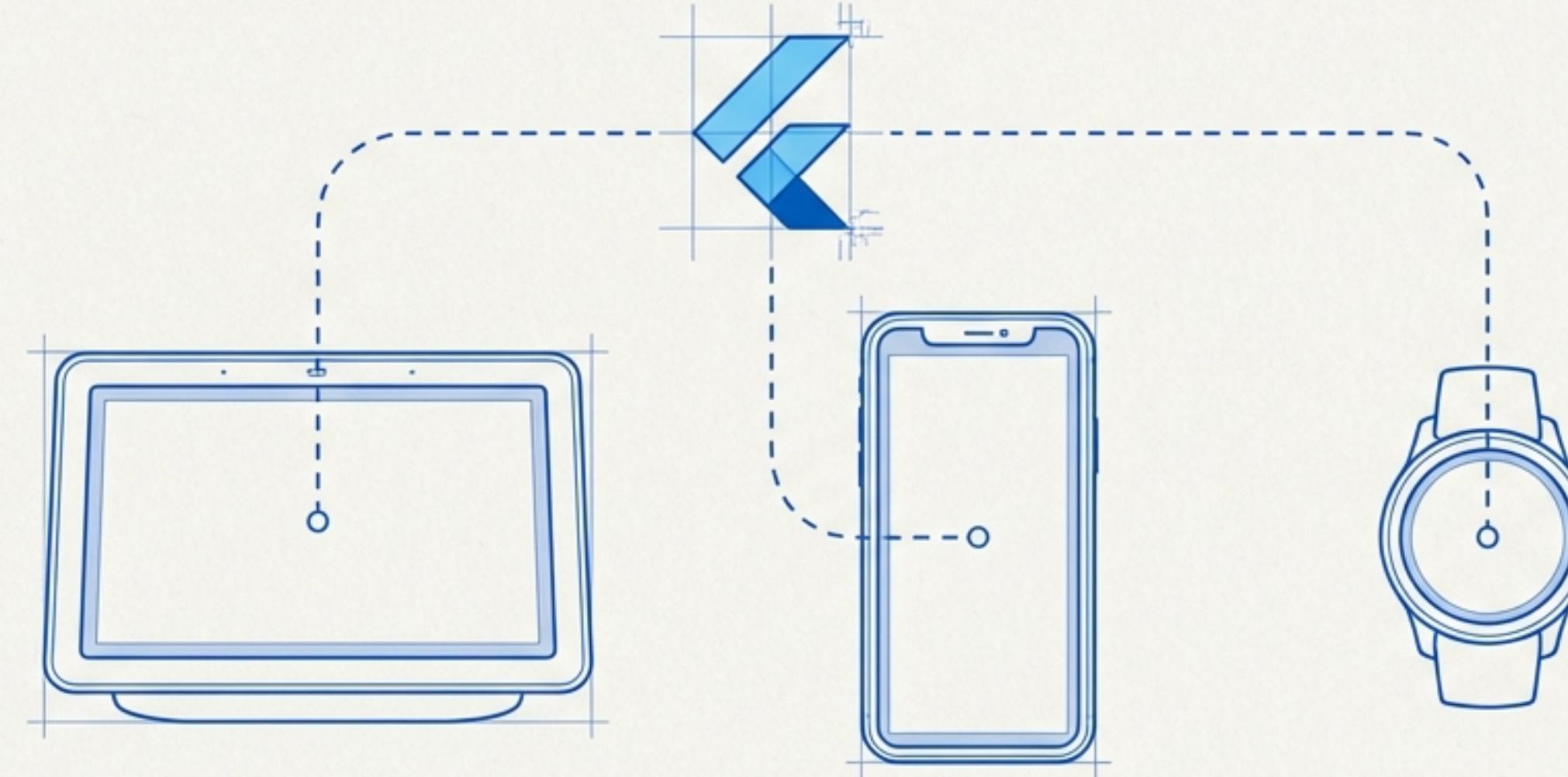
2016-Future



Interface e Aplicações: O Papel do Flutter

UI Toolkit:
Flutter (Dart)

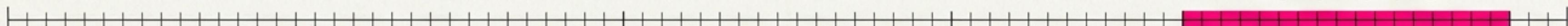
Motor Gráfico:
Escher (Vulkan)



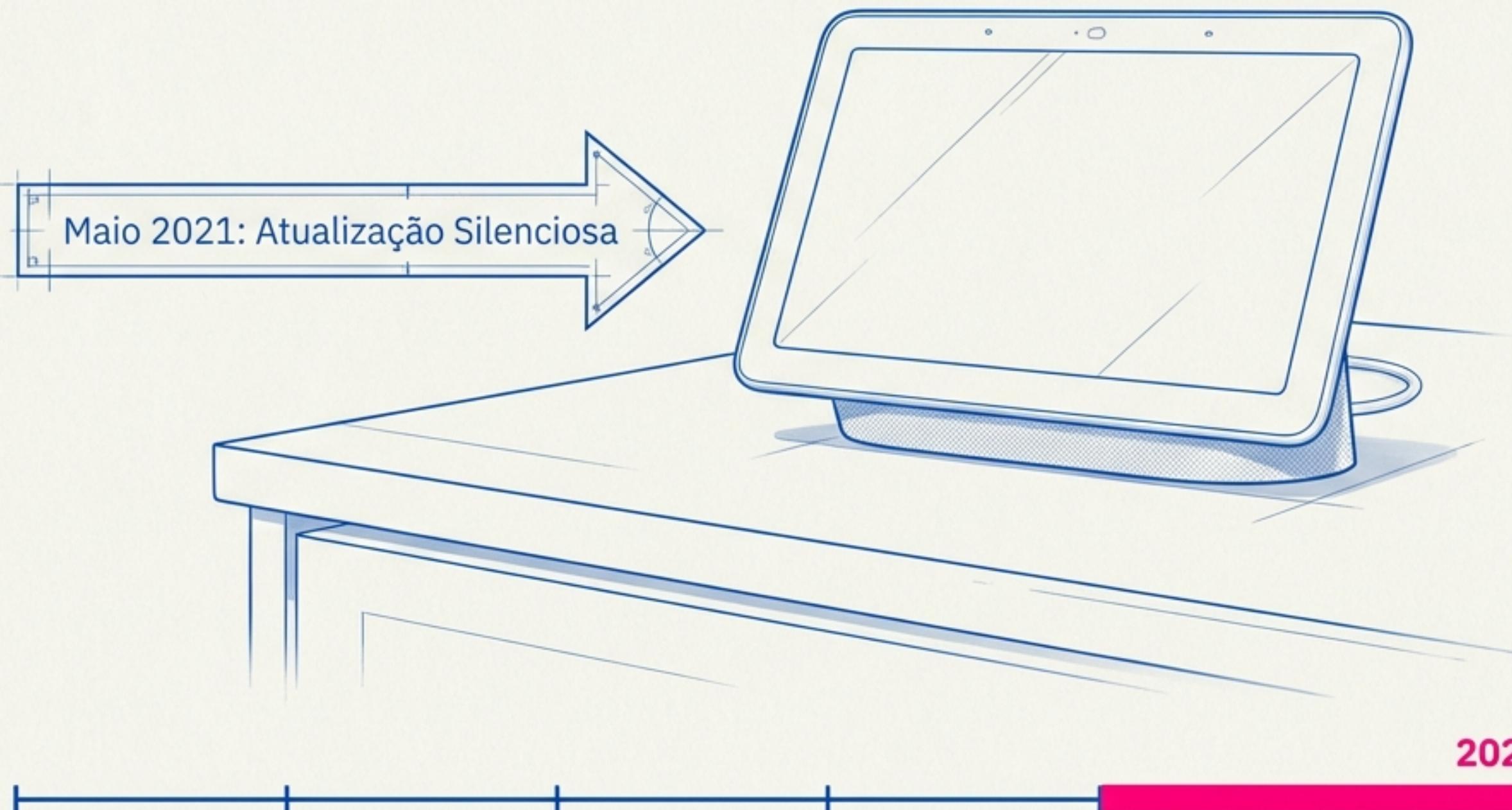
**Insight: Desacoplamento total: A interface é independente
do sistema operacional.**

Suporte a Apps Android via Runtime (.far)

2016-Future



Fuchsia no Mundo Real



Google substituiu o Cast OS (Linux) pelo Fuchsia em milhões de dispositivos via update OTA.

Insight: Nenhuma mudança visual para o usuário. Uma troca de motor com o carro em movimento.

Comparativo de Gigantes

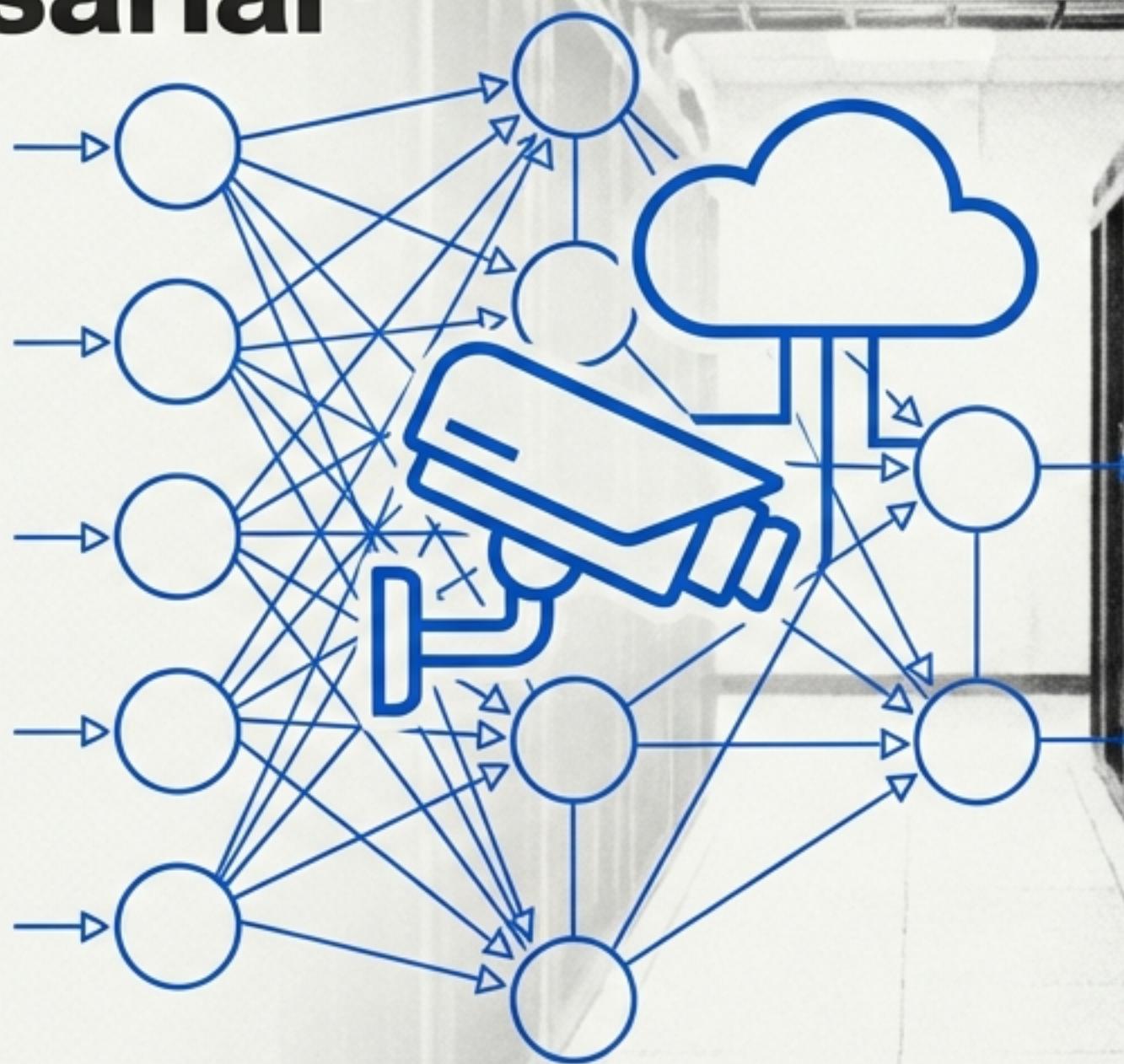
LINUX (Android)	XNU (macOS/iOS)	ZIRCON (Fuchsia)
Kernel Monolítico	Kernel Híbrido	Microkernel
Licença GPL	Licença Mista	Licença BSD/MIT
Foco: Geral/Servidor	Foco: Consumer	Foco: IoT/Real-time

O Futuro: Sistemas de Inteligência Empresarial

A próxima fronteira não é apenas rodar apps, mas processar o mundo.

Nx EVOS
Gerenciamento de IA e Vídeo

Insight: Do OS de Computação Geral para o OS de Decisão.



Resumo e Reflexão: A História da Abstração

