

# Clonezilla: Efficient Deployment & Rescue for x86, ARM, and RISC-V

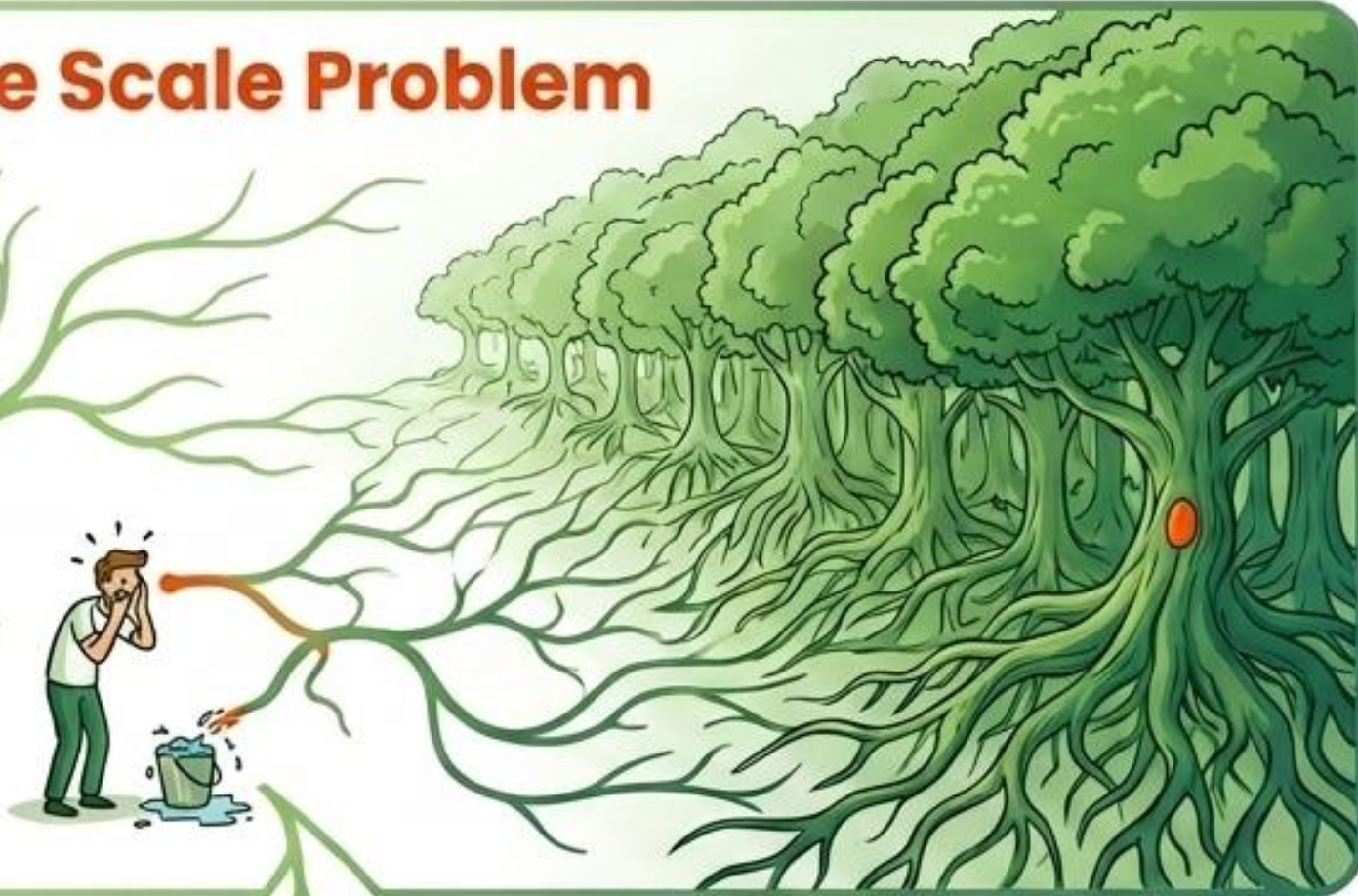
Steven Shiau, Ceasar Sun, Thomas Tsai,  
LFNW 2026



Clonezilla

# The Bare-Metal Deployment Nightmare

## The Scale Problem



- Provisioning labs or clusters 1-by-1 is a massive time sink.
- Proprietary ghosting software requires expensive, restrictive licensing.

## The Crash Problem

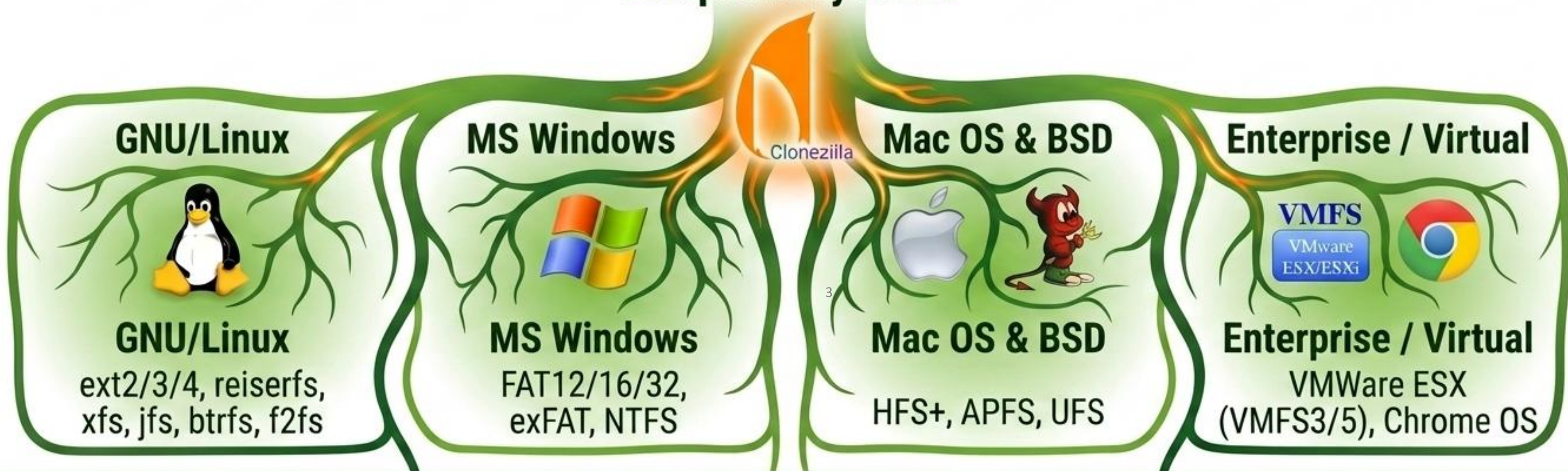


- Catastrophic **hardware** failures demand immediate bare-metal restoration.
- Traditional sector-by-sector copying copies empty space, wasting hours and terabytes.

# The Open-Source Swiss Army Knife

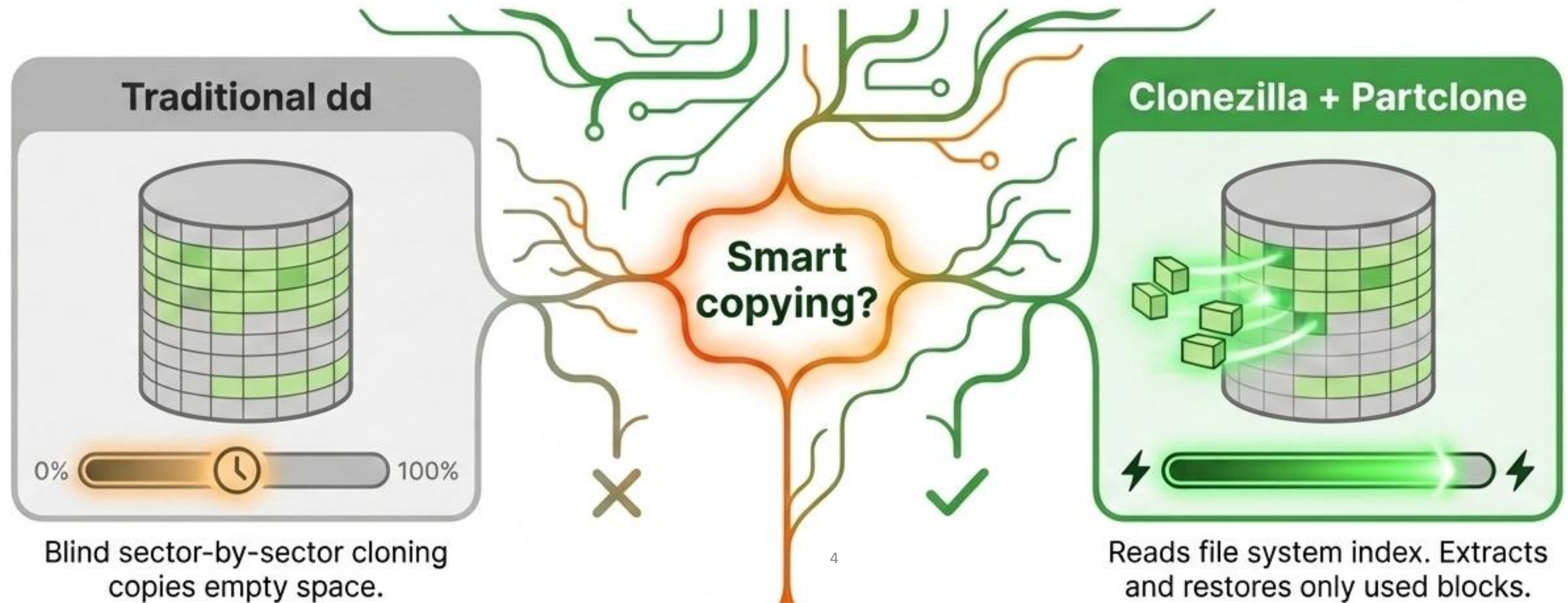
Since 2007, the true GPL-licensed alternative for system deployment, bare-metal backup, and recovery.

## Compatibility Roots



**Security:** Native support for LUKS encrypted partitions and ecryptfs image encryption

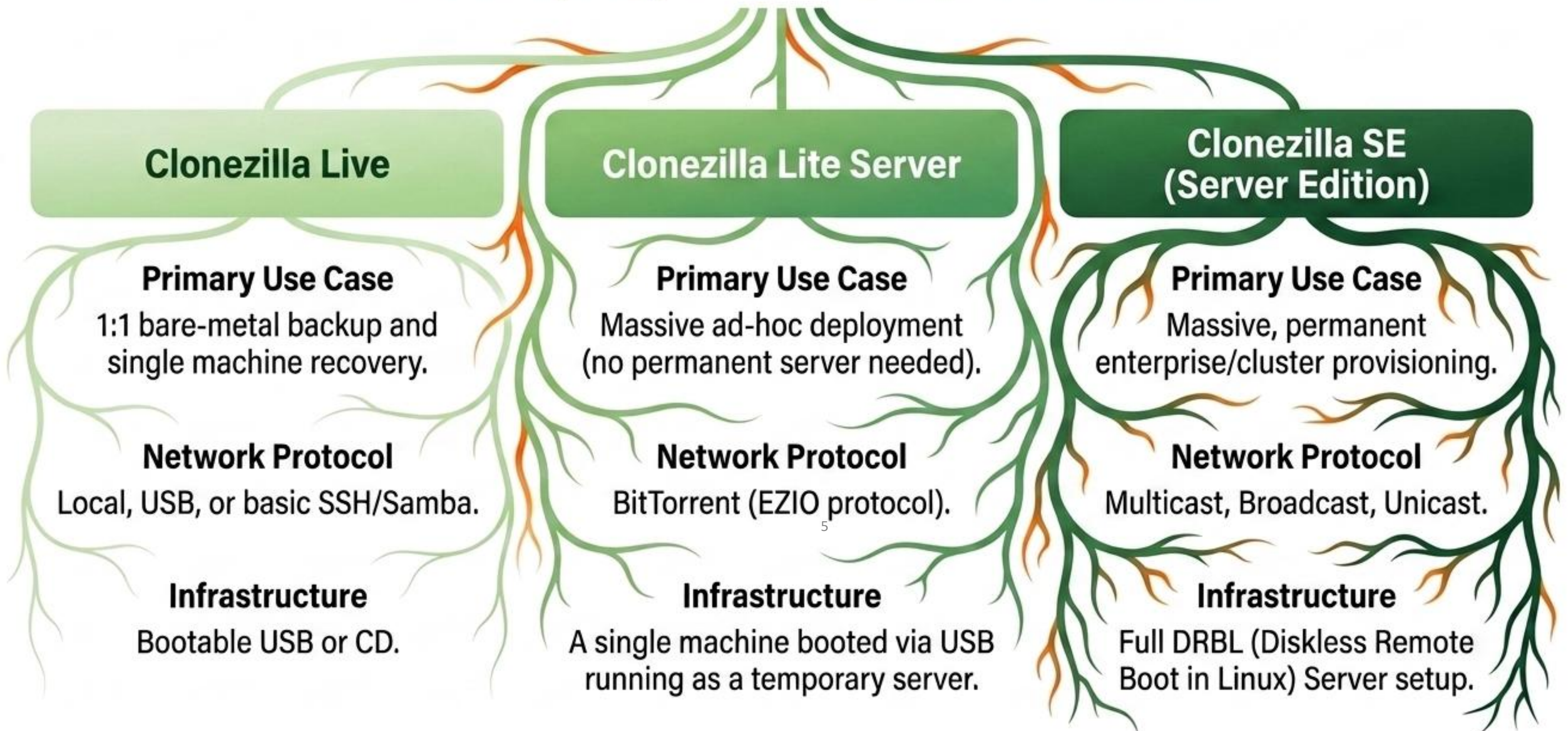
# The Engine: Smart Block Copying



## Pipeline Efficiency

These blocks are instantly piped through sequential or multi-core compression engines (gzip, pigz, xz, pbzip2) to minimize storage footprint on the fly.

# The Deployment Scale Matrix



# Massive Scale: Multicast Deployment

## The Mechanics

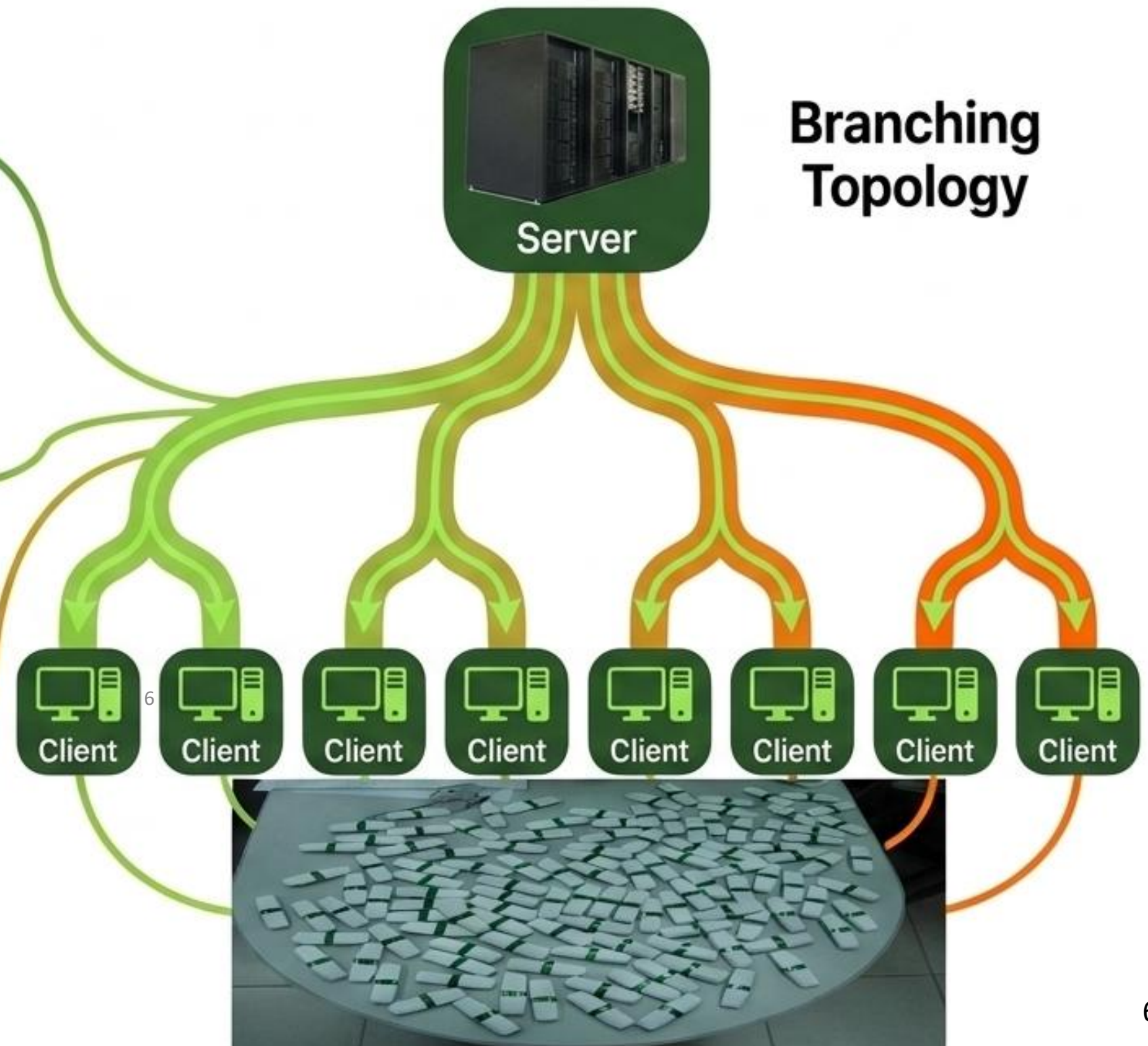
Clonezilla SE utilizes UDPcast and PXE boot (Wake-on-LAN) to broadcast a single image to multiple clients simultaneously.

## The "Clients-to-Wait" Orchestration

The server prepares the PXE kernel and acts as a TFTP server, holding the image stream until a specified number of clients connect, then blasts the data to all at once.

## Benchmark

Proven to clone 40+ computers simultaneously. With high-end hardware in a 42-node cluster, multicast restoring hit rates of **8 GB/min.**



# Massive Scale: BitTorrent Provisioning

## The Bottleneck Problem

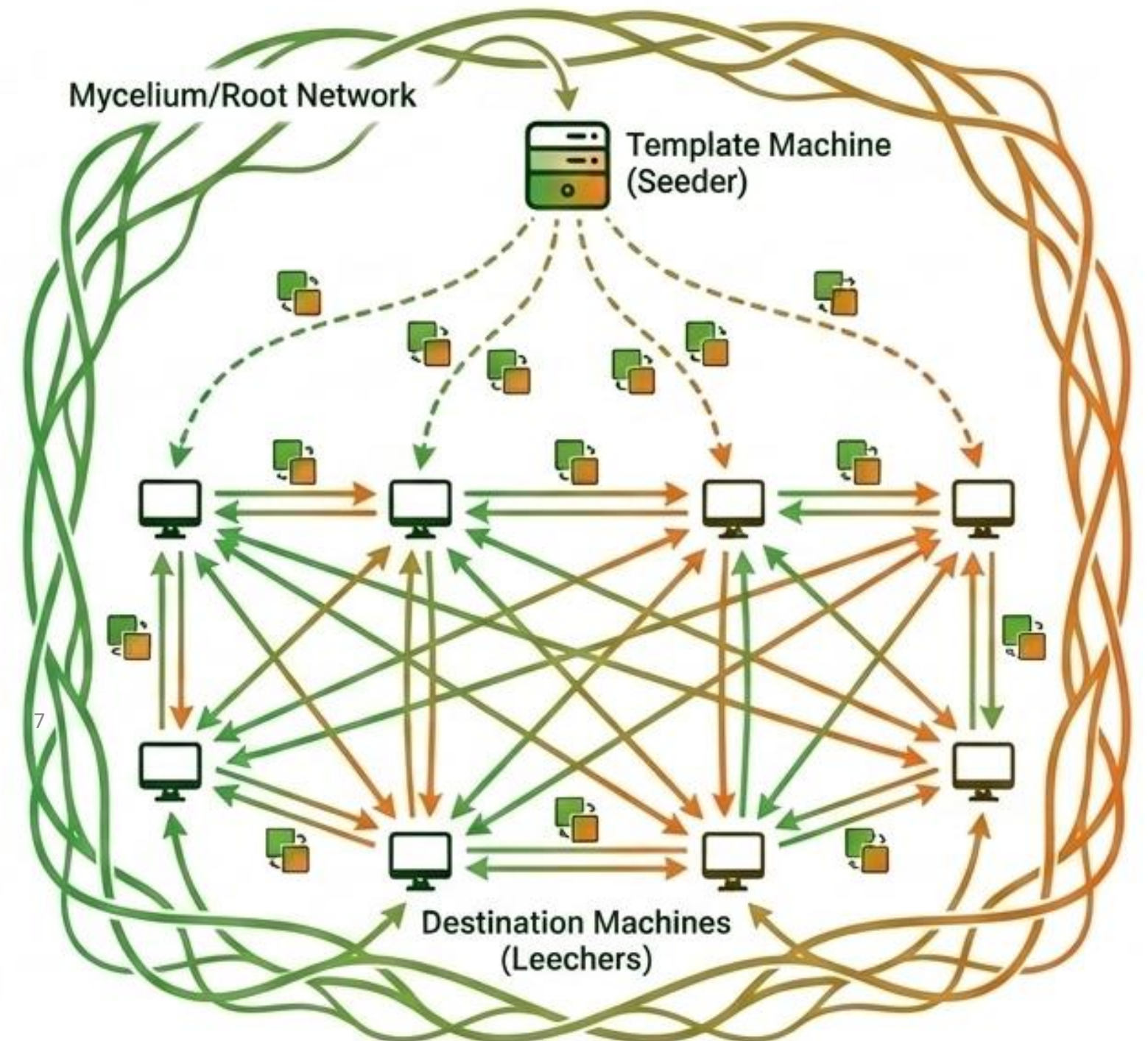
Traditional server-client models choke the central server's network interface during massive deployments.

## The EZIO Solution

Clonezilla Lite Server integrates the EZIO BitTorrent protocol. EZIO reads only used blocks from the source template machine (seeder). As destination machines (leechers) receive blocks, they simultaneously write to their drives and seed those blocks to peer machines.

## Result

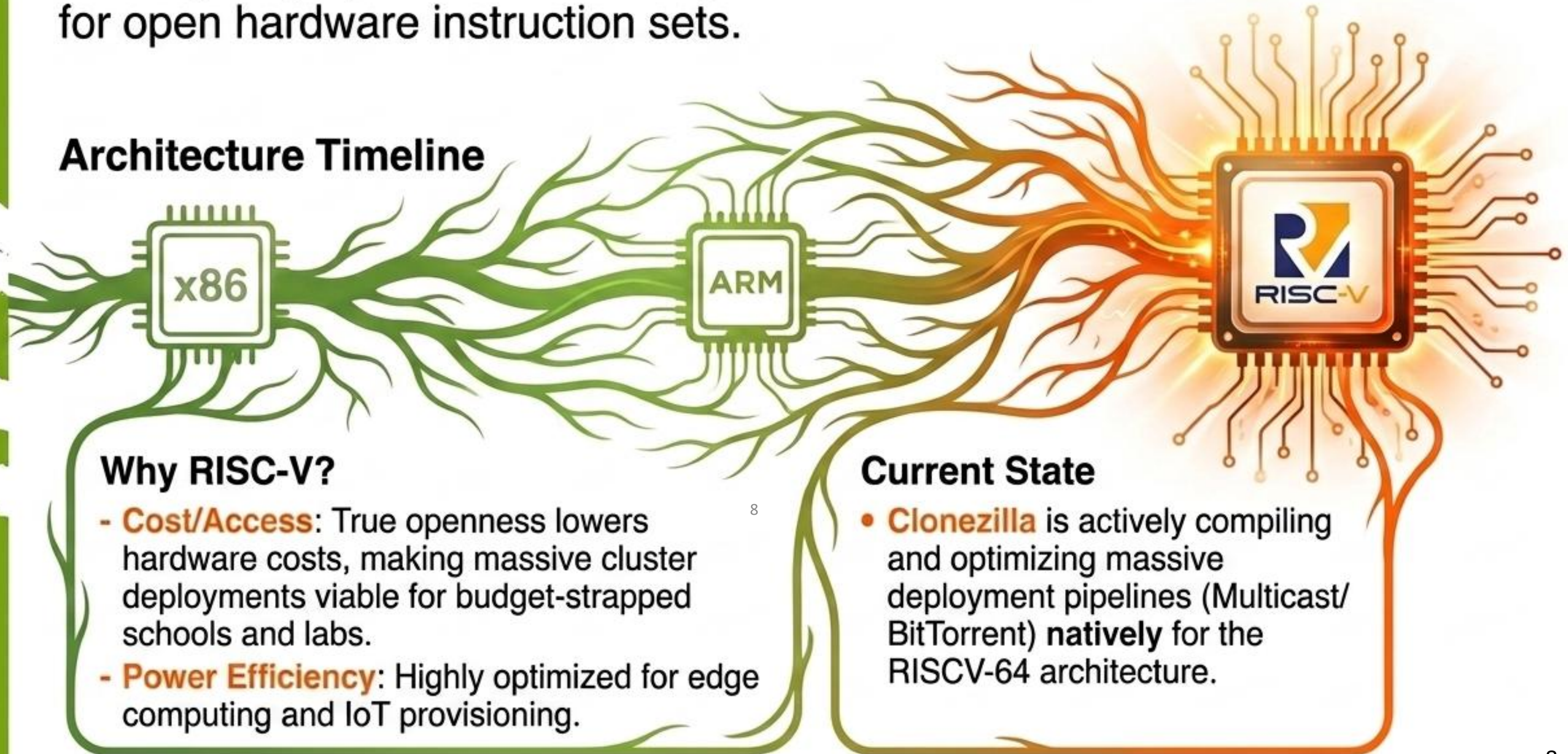
The network load is decentralized. The more machines you deploy to, the stronger and faster the network becomes.



# Cross-Architecture Evolution: The RISC-V Frontier

Crafting fully **open-source** live systems tailored for open hardware instruction sets.

## Architecture Timeline



### Why RISC-V?

- **Cost/Access:** True openness lowers hardware costs, making massive cluster deployments viable for budget-strapped schools and labs.
- **Power Efficiency:** Highly optimized for edge computing and IoT provisioning.

### Current State

- **Clonezilla** is actively compiling and optimizing massive deployment pipelines (Multicast/BitTorrent) **natively** for the RISC-V-64 architecture.

# The Clonezilla Ecosystem Stack

## Deployment Orchestrators

UDPcast (Multicast), Ezio (BitTorrent),  
PXE/TFTP bootloaders.

## Engine & Tooling

Partclone (smart block extraction),  
sgdisk (partition tables), DRBL-Winroll  
(Windows SID/Hostname automation).

## Base OS

Debian/Ubuntu Live environments  
providing the fundamental Linux  
kernel and hardware drivers.

## Open Hardware

x86, ARM, RISC-V

### Synthesis Insight

Clonezilla isn't just an app; it is an orchestration layer that binds the best open-source GNU/Linux tools into a unified, Enterprise-grade deployment engine.

# Come Talk to the Roots.

## Visit the Clonezilla Booth at LinuxFest Northwest 2026.

What to expect:

- See the live **RISC-V (RISCV-64)** deployment demo.
- Discuss your specific cluster topology (Multicast vs. BitTorrent) with the core maintainers.
- Explore custom **Clonezilla** for your lab or enterprise.

 **clonezilla.org**