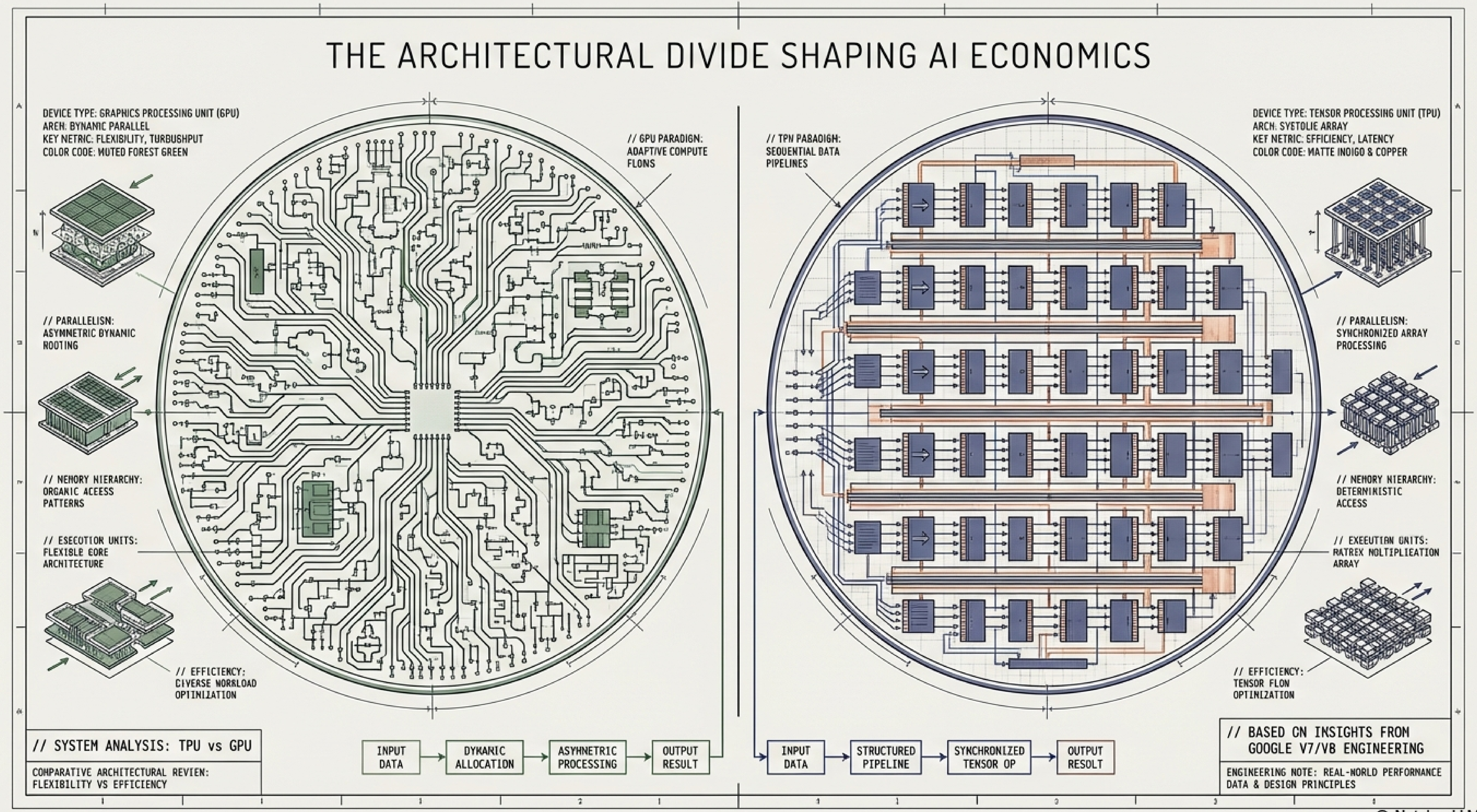
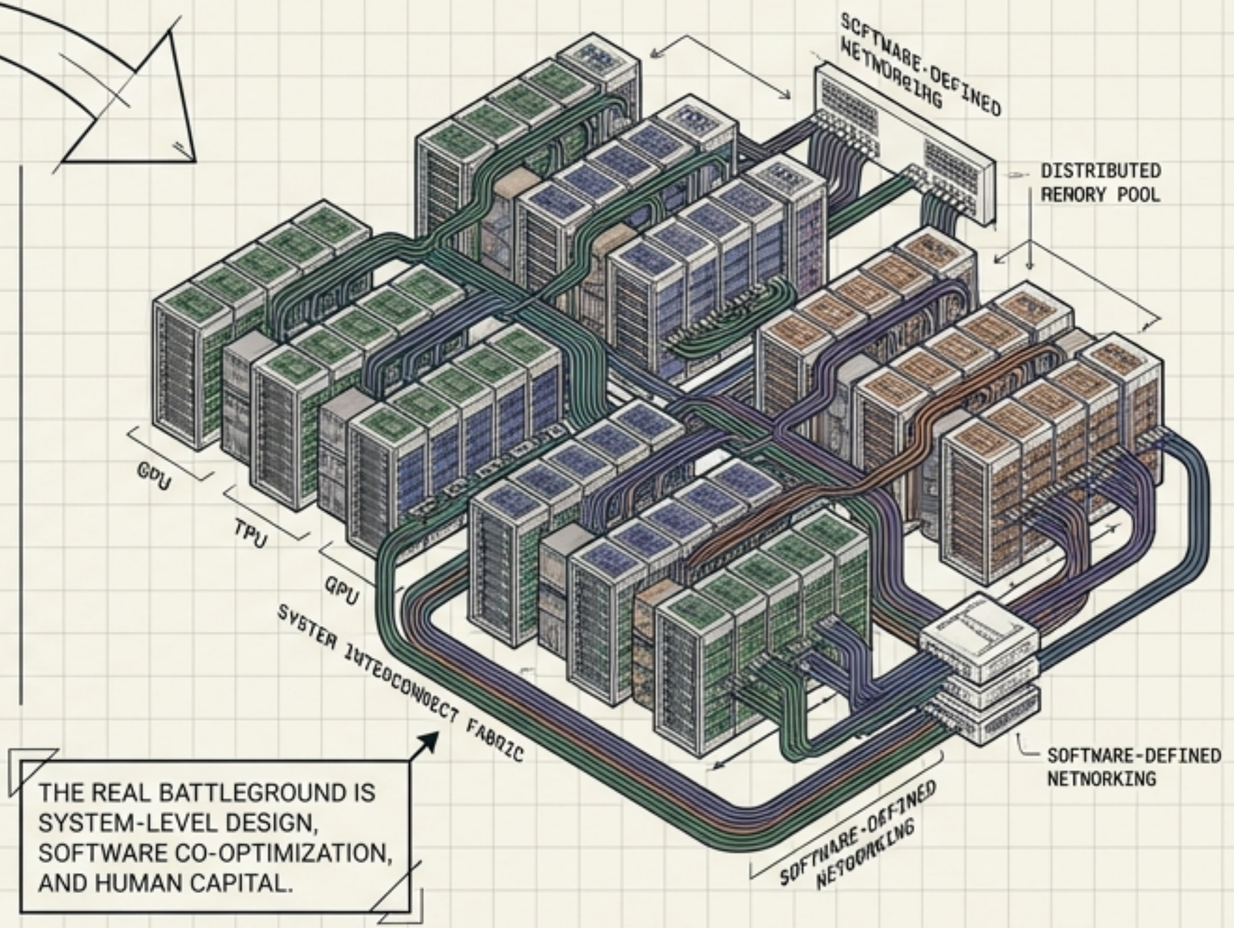


THE ARCHITECTURAL DIVIDE SHAPING AI ECONOMICS
DEVICE TYPE: GRAPHICS PROCESSING UNIT (GPU)
AREH: BYNANIC PARALLEL
KEY NETRIC: FLEXIBILITY, TURBUGHPUT
COLOR CODE: MUTED FOREST GREEN
// GPU PARADIGN: ADAPTIVE COMPUTE FLONS
// PARALLELISM: ASYMNETRIC BYNANIC ROOTING
// NEMORY NIERARCHY: ORGANIC ACEESS PATTERRS
// ESECOTION UNITS: FLEXIBLE EORE ARCHITEETURE
// EFFICIENCY: CIVERSE NORKLOAD OPTIMIZATION
DEVICE TYPE: TENSOR PROCESSING UNIT (TPU)
ARCH: SYSTOLIE ARRAY
KEF NETRIC: EFFICIENCY, LATENCY
COLOR CODE: MATTE INOIGO & COPPER
// TPN PARADIGH: SEQUENTIAL DATA PIPELINES
// PARALLELISN: SYNCHRONIZED ARRAY PROCESSING
// NEMORY NIERARCHY: DETERMINISTIC ACCESS
// EXEEUTIAN UNITS: RATREX NOLTIPLIEATION ARRAY
// EFFICIENCY: TENSOR FLON OPTIMIZATION
// SYSTEM ANALYSIS: TPU vs GPU
COMPARATIVE ARCHITECTURAL REVIEN: FLEXIBILITY VS EFFICIENCY
INPUT DATA
DYKANIC ALLOCATION
ASYHNETRIC PROCESSING
OUTPUT RESULT
INPUT DATA
STRUCTURED PIPELINE
SYNCHRONIZED TENSOR OP
OUTPUT RESULT
// BASED CN INSIGHTS FROM GOOGLE V7/V8 ENGINEERING
ENGINEERING NOTE: REAL-NORLD PERFORMANCE DATA & DESIGN PRINCIPLES
NotebookLM

THE OLD MENTAL MODEL FOCUSES ON RAW COMPUTE.

SPECIFICATION	VALUE
TERAFLOPS	100 TF
MEMORY BANDWIDTH	1 TB/S
CORE COUNT	8192 CORES

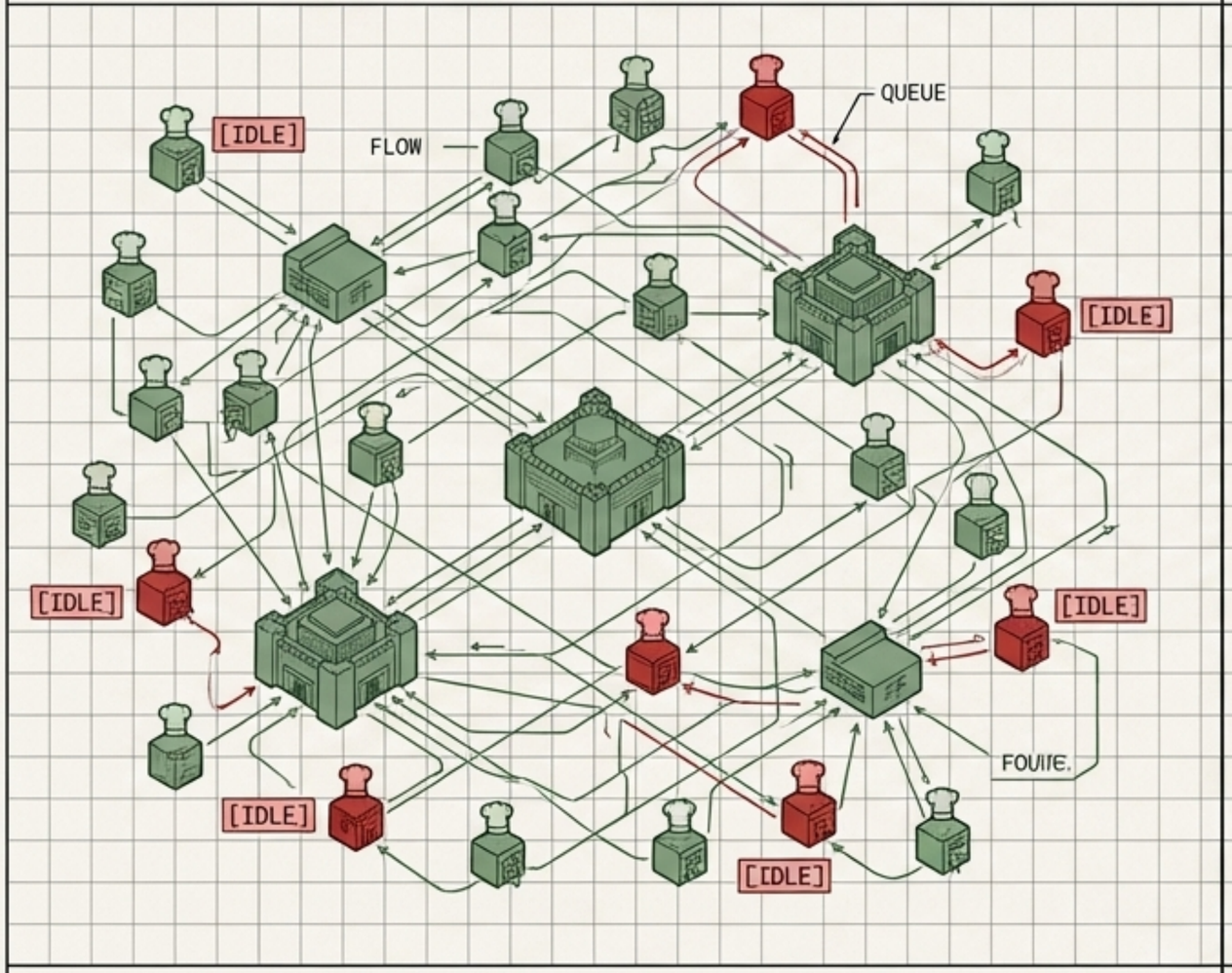


THE REAL BATTLEGROUND IS SYSTEM-LEVEL DESIGN, SOFTWARE CO-OPTIMIZATION, AND HUMAN CAPITAL.

THE HARDWARE MATTERS LESS THAN WHERE THE SYSTEM PLACES ITS INTELLIGENCE.

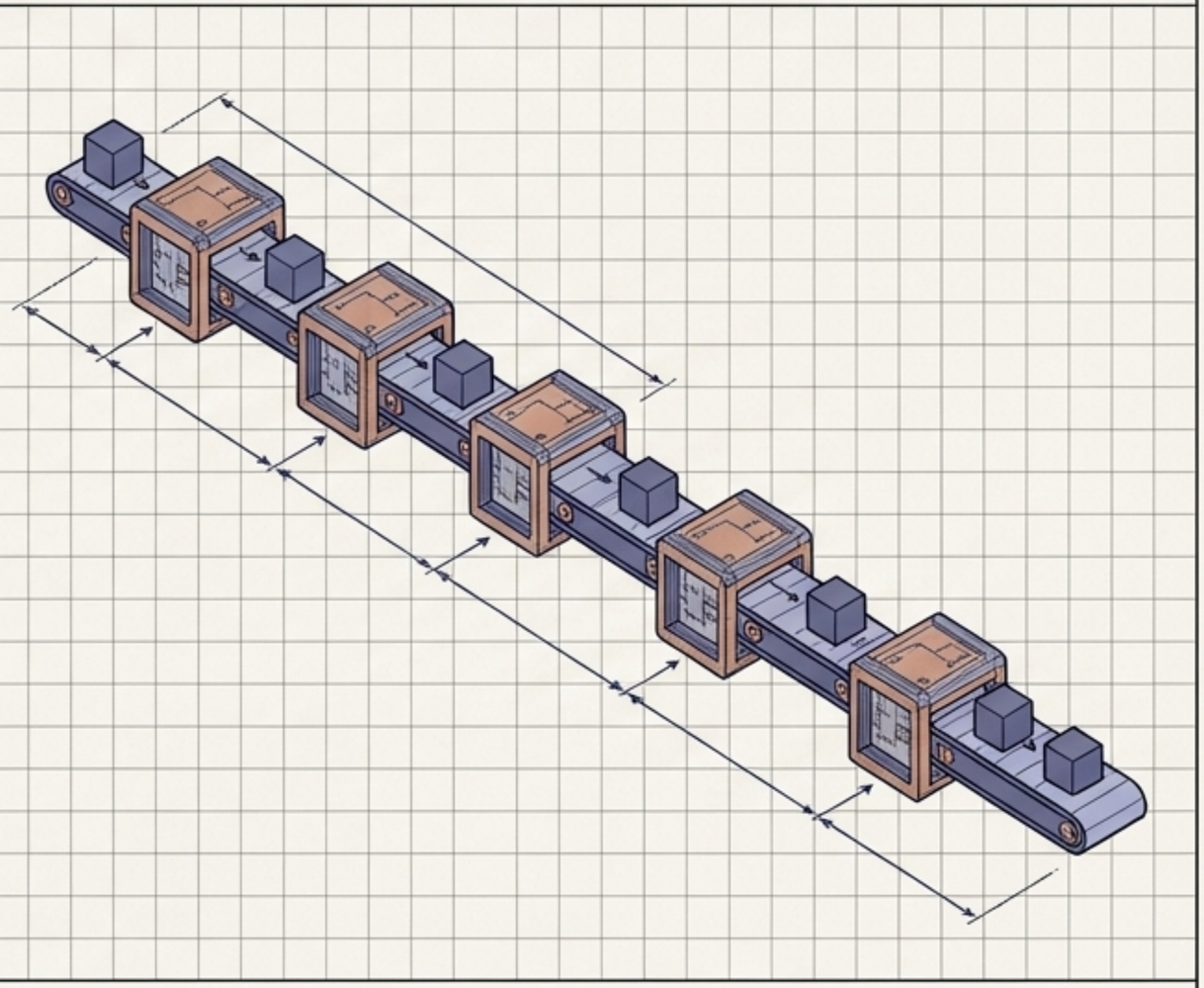
Execution Philosophies: SIMT vs. XLA

The Independent Chefs (SIMT)



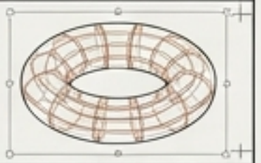
Single Instruction Multiple Threading. Massive parallelism, but inherent runtime friction and idle waiting.

The Assembly Line (XLA)

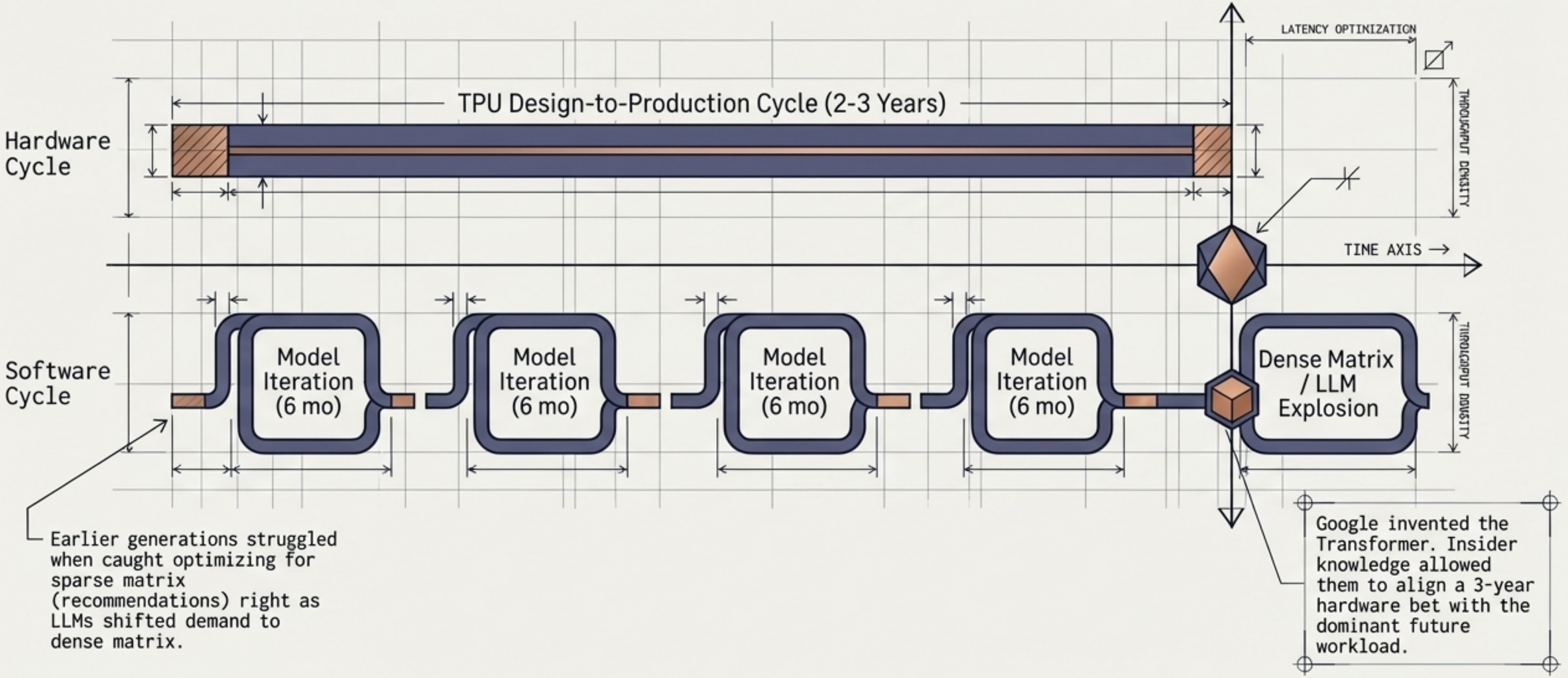


Omniscient ahead-of-time compilation. The software schedules every clock cycle. The hardware executes mechanically.

The Paradigm Divide

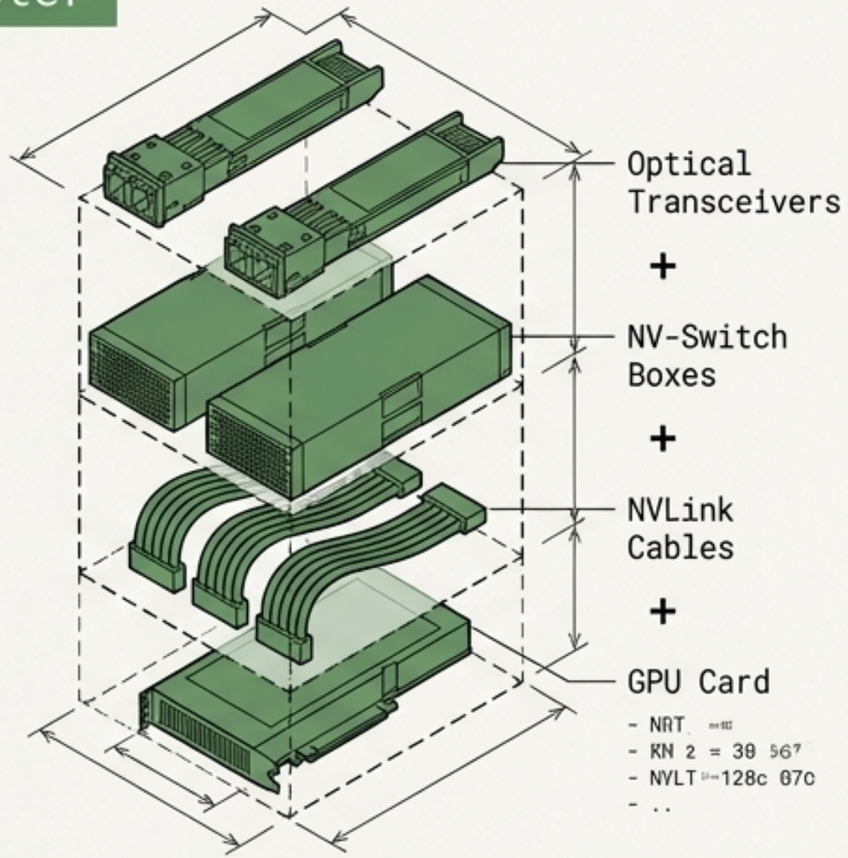
	GPU Architecture	TPU Architecture
Core Philosophy	Smart Hardware (Dynamic runtime prediction)	Dumb Hardware + Smart Software
Execution Metaphor	Independent Chefs	Flawless Assembly Line
Networking	Infrastructure Tax (External optical switches)	Native 3D Torus (Direct copper interconnects) 
Ecosystem	Open & Community-driven (CUDA / PyTorch)	Closed & Talent-dependent (JAX / XLA)
Adaptability	High (Adapts via software updates)	Low (Requires 2-3 year hardware bets)
Defect Tolerance	High (Binning and downgrading works)	Zero (System coherence requires flawless yield)

The ASIC Bet: High Reward, Existential Risk



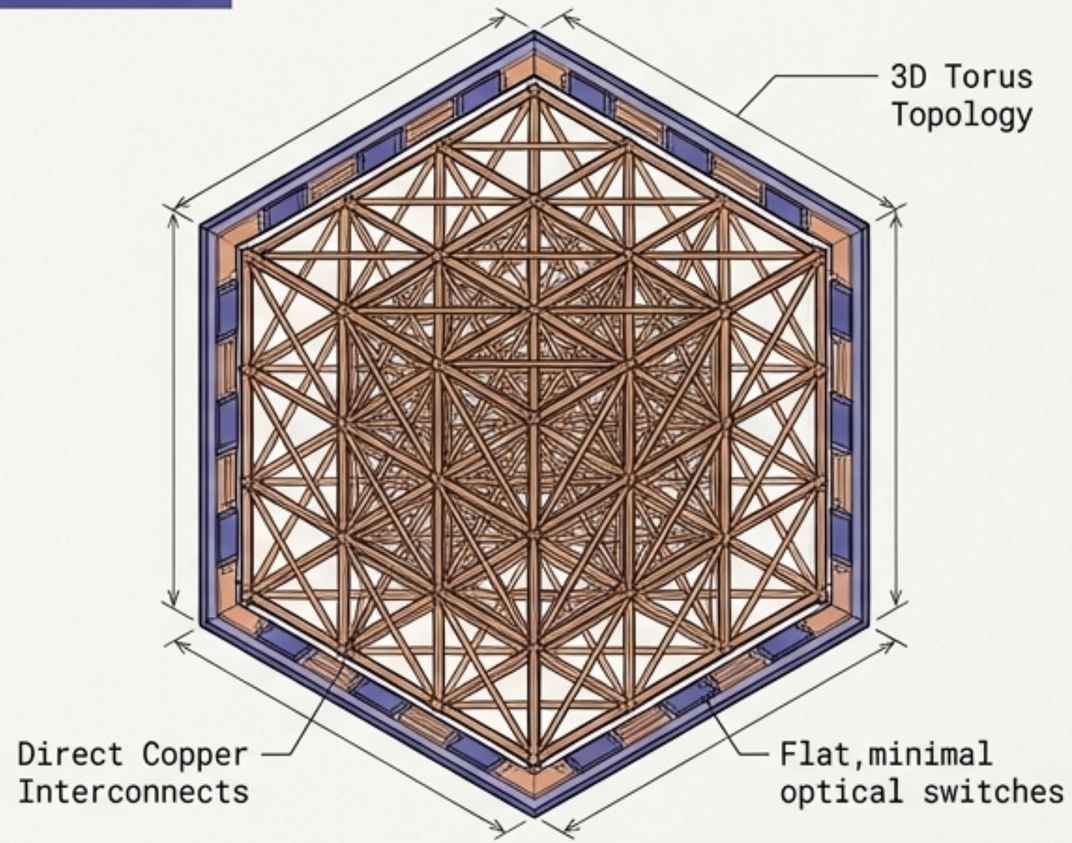
Scaling Up: The Infrastructure Tax

GPU Cluster



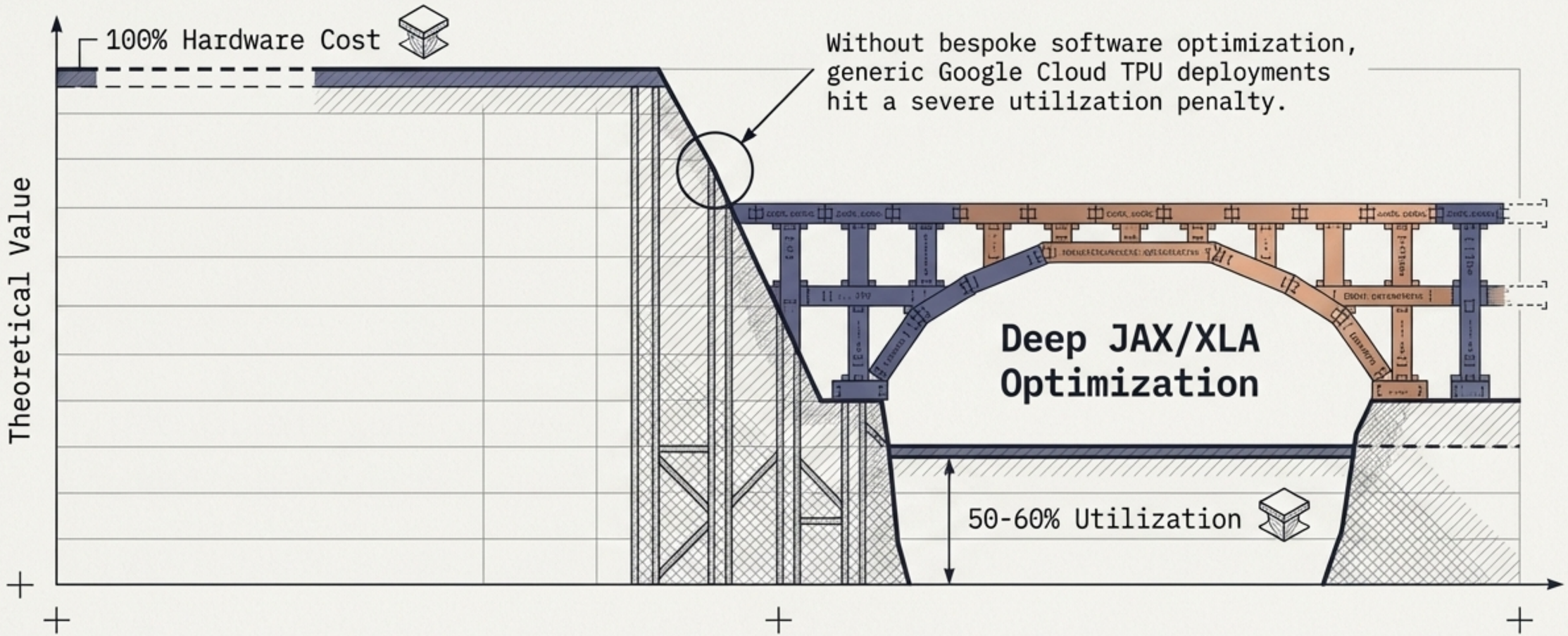
Scaling requires buying expensive, separate networking hardware to tie individual cards together.

TPU Pod



Designed as a cluster from Day 1. Software-configurable routing eliminates intermediate switch costs for training large models.

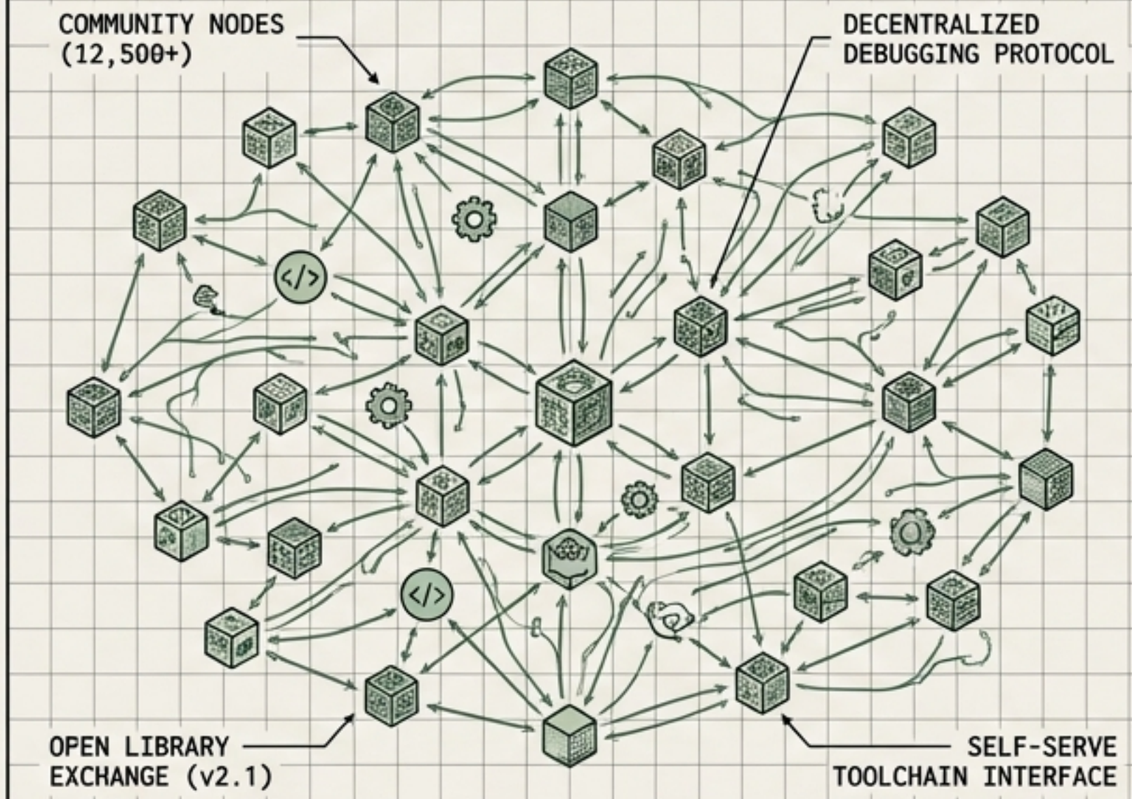
The Utilization Cliff



Paying for 100% of the hardware while extracting 50% of the compute destroys the TCO advantage.

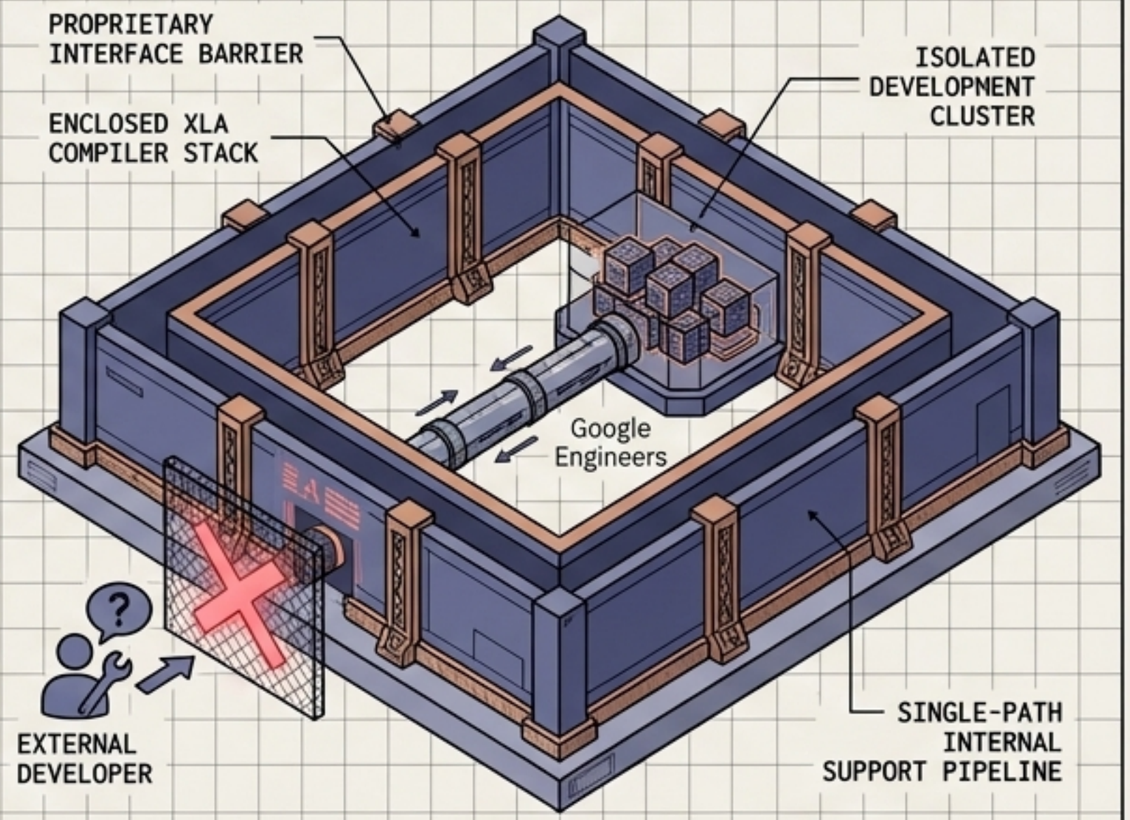
The Developer Moat: Open Commons vs. The Closed Compiler

Open Commons (CUDA/PyTorch)



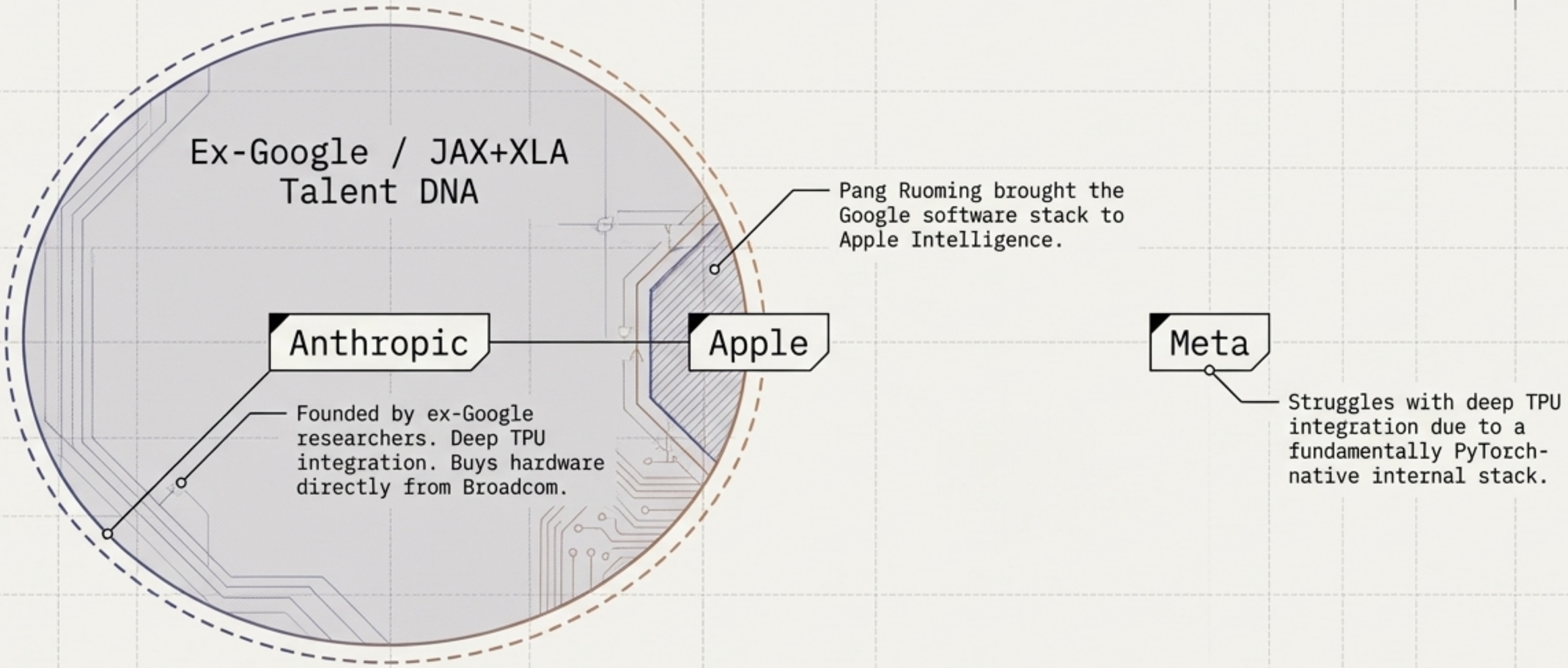
The Lingua Franca.
Massive community, open toolchains, self-serve debugging.

The Closed Compiler (JAX/XLA)



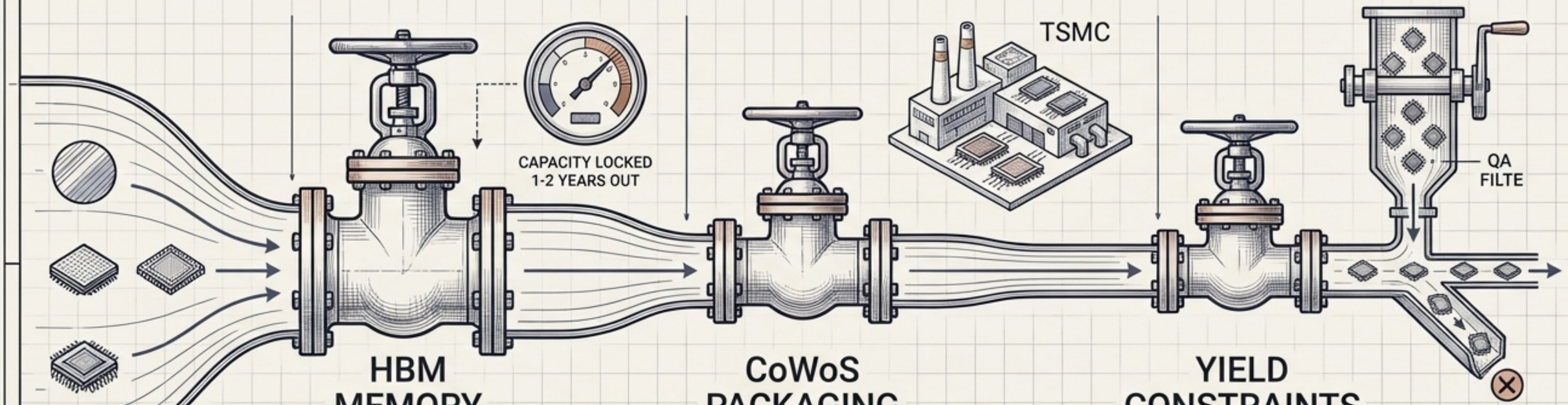
Static compiler perfection requires a steep trade-off: external developers cannot independently fix bugs. They rely entirely on internal Google support.

The Ex-Google Talent Map



TPU’s value proposition requires human capital that is intensely scarce and highly concentrated.

The Physical Supply Chain Funnel



HBM MEMORY

Nvidia is the priority volume customer. Capacity locked 1-2 years out.

CoWoS PACKAGING

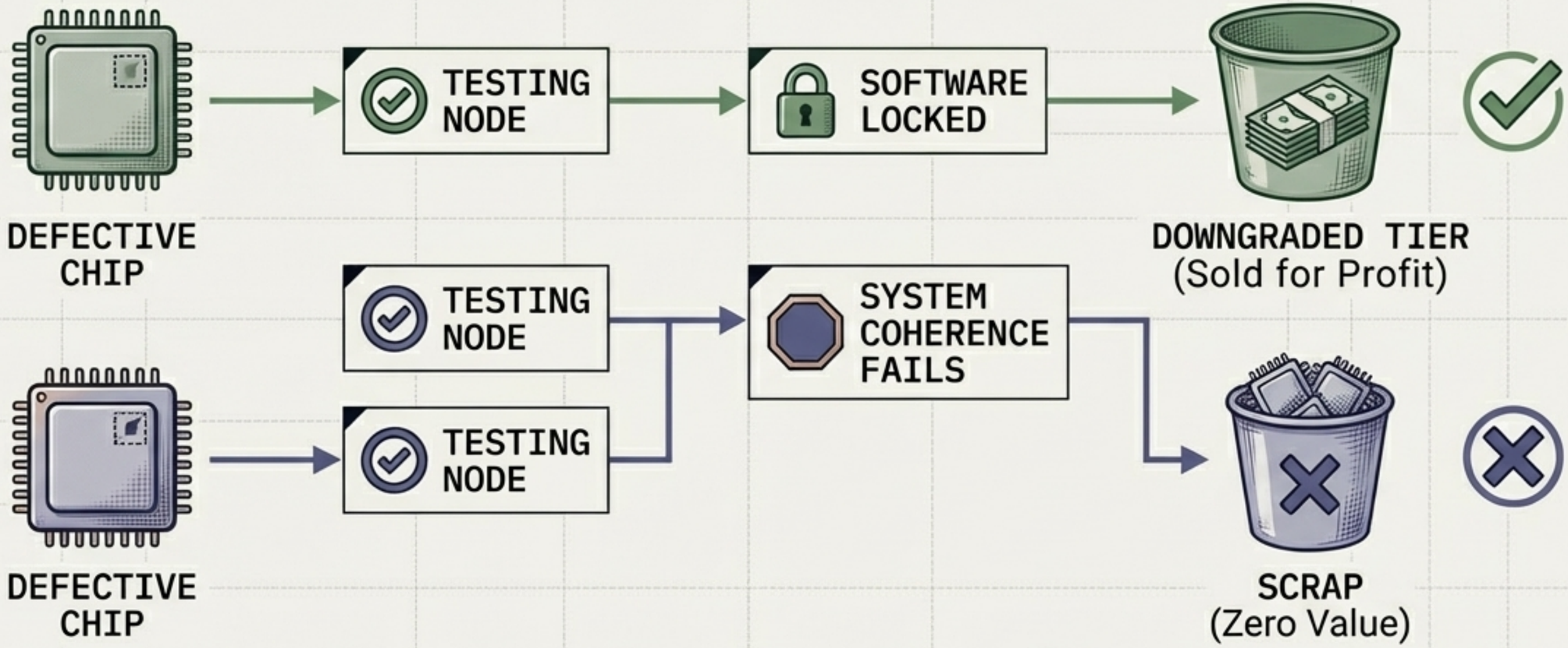
TSMC allocates advanced packaging by order volume. Nvidia dominates allocation.

YIELD CONSTRAINTS

Pod-level coherence requires flawless chips. Cannot bin or downgrade.

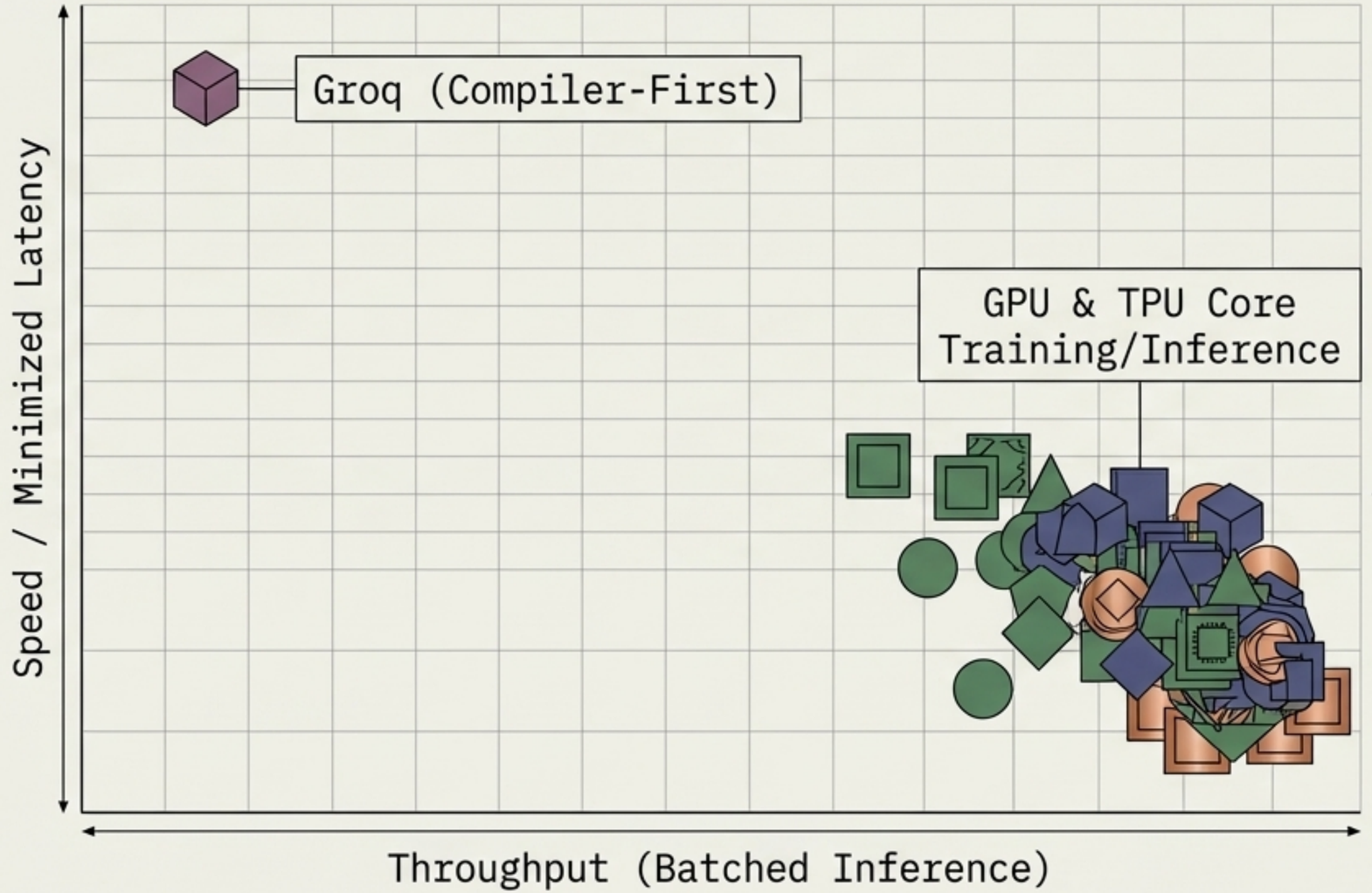
Three critical manufacturing chokepoints—all entirely outside of Google's control.

The Economics of Yield: Binning vs. Scrap



GPU architectures recover manufacturing costs through downgrading. TPU architecture demands perfection, directly constraining volume and margins.

Groq: The Compiler-First Extreme



Founded by Google TPU compiler alumni.

Hardware is even simpler than TPU; the compiler controls individual clock cycles with zero runtime decisions.

Trades raw throughput for sub-millisecond response.

Purpose-built for the agent era where multi-step logic chains compound latency.

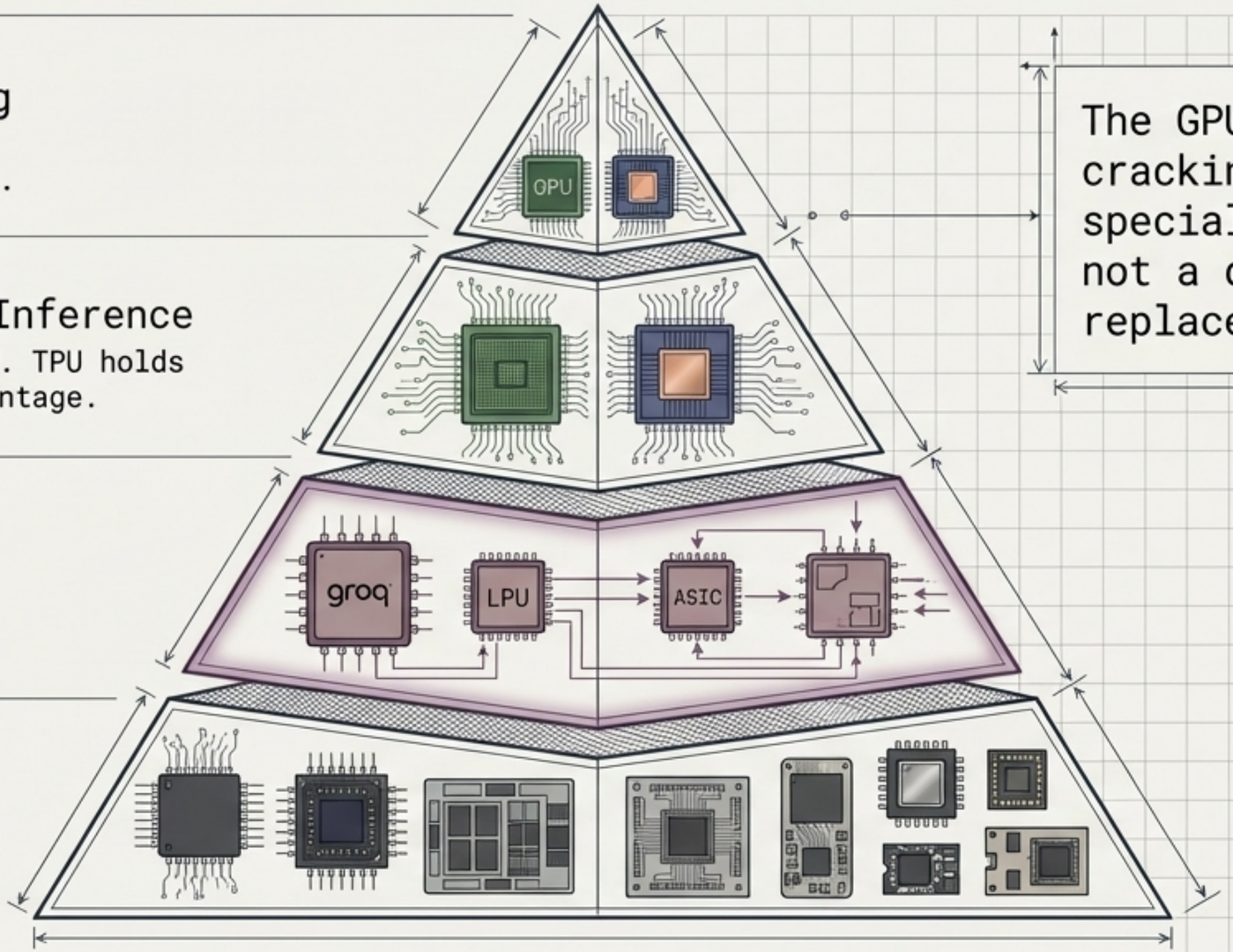
The Stratification of AI Hardware

Frontier Training
Heaviest workload.
Populated by GPU & TPU.

High-Throughput Inference
Populated by GPU & TPU. TPU holds theoretical scale advantage.

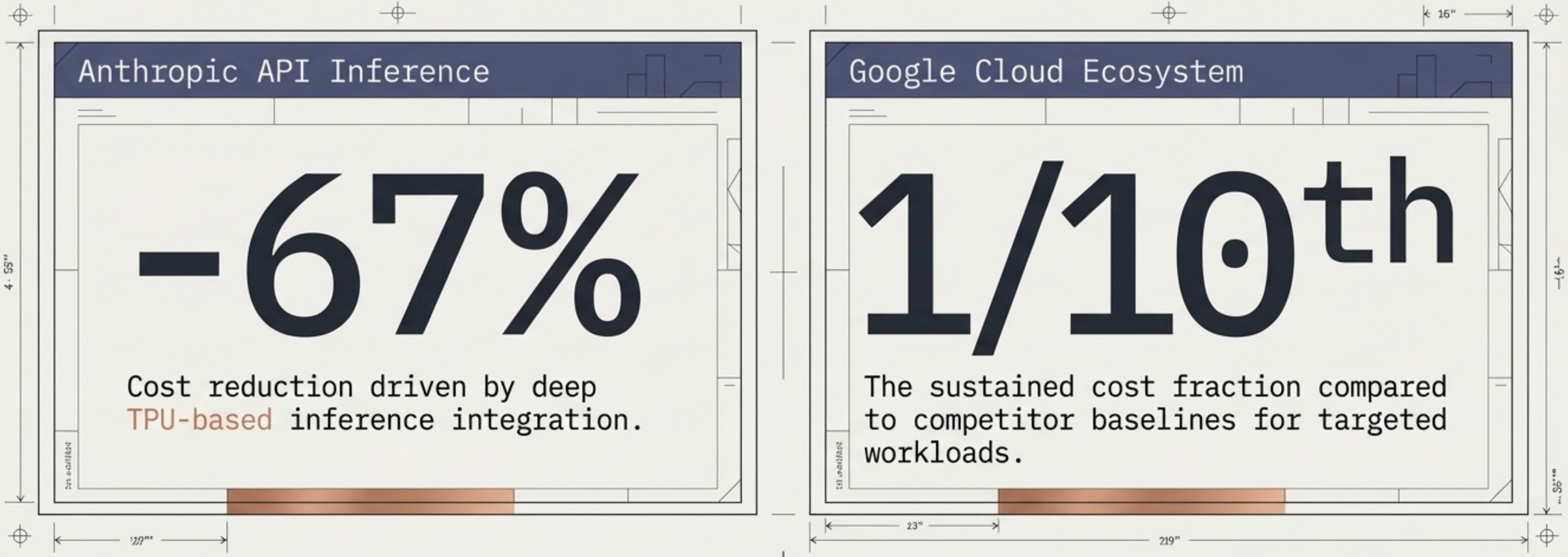
Low-Latency / Agent Inference
Populated by Groq & Latency ASICs.

Edge / Local Deployment
Highly diverse, specialized silicon.



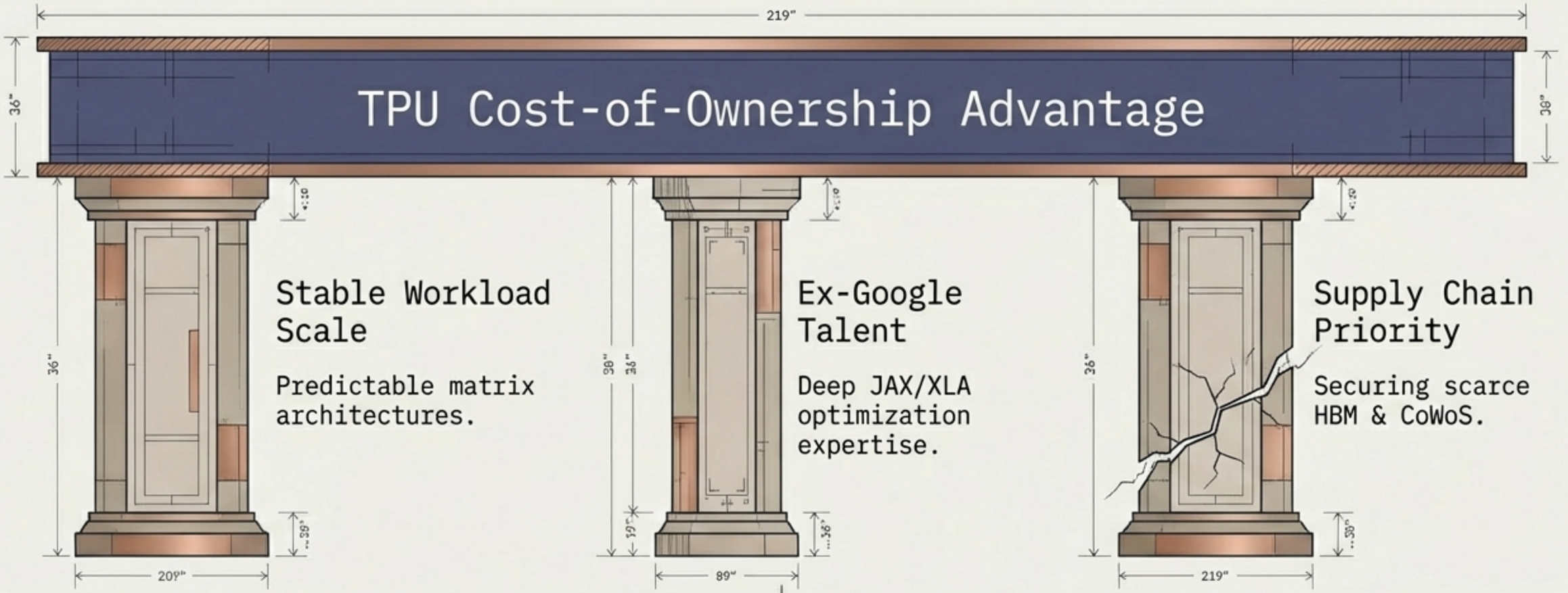
The GPU monopoly is cracking, but into specialized layers, not a direct 1:1 replacement.

Downstream Economics: The API Pricing Effect



- Even if you never train a model, the TPU vs. GPU battle impacts you.
- For consumers of AI APIs, structural inference advantages translate directly to aggressive price compression.

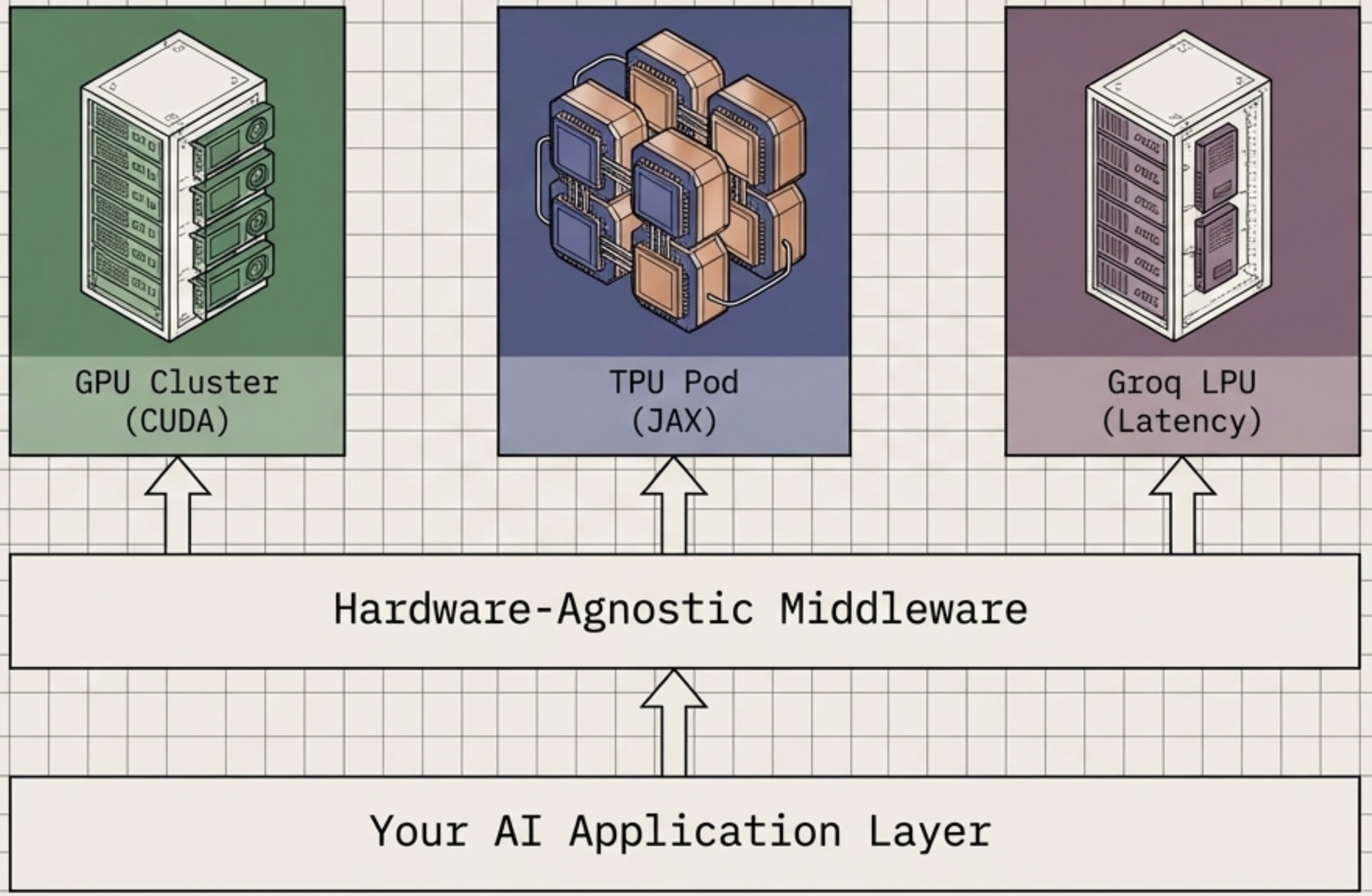
Synthesis: The Fragile Juggernaut



- TPU's theoretical dominance is not plug-and-play. It requires perfect alignment of these three massive dependencies.
- If any single pillar fails, the GPU wins by default via sheer flexibility.

Strategic Imperative for Builders

Do not hardcode your company's future into a single platform's assumptions.



The defining trait of surviving companies won't be picking the right hardware.

It will be building hardware-agnostic software layers capable of shifting between architectures as economics, capabilities, and supply chains evolve.