Confidential Compute in Reality

Bertram Dorn

Principal, Office of the CISO
AWS Security

Economics

The Attacker needs to win once,

the Defender needs to win always

The Attacker knows when he has won something,

the Defender might never know

How can this Geometry be changed?

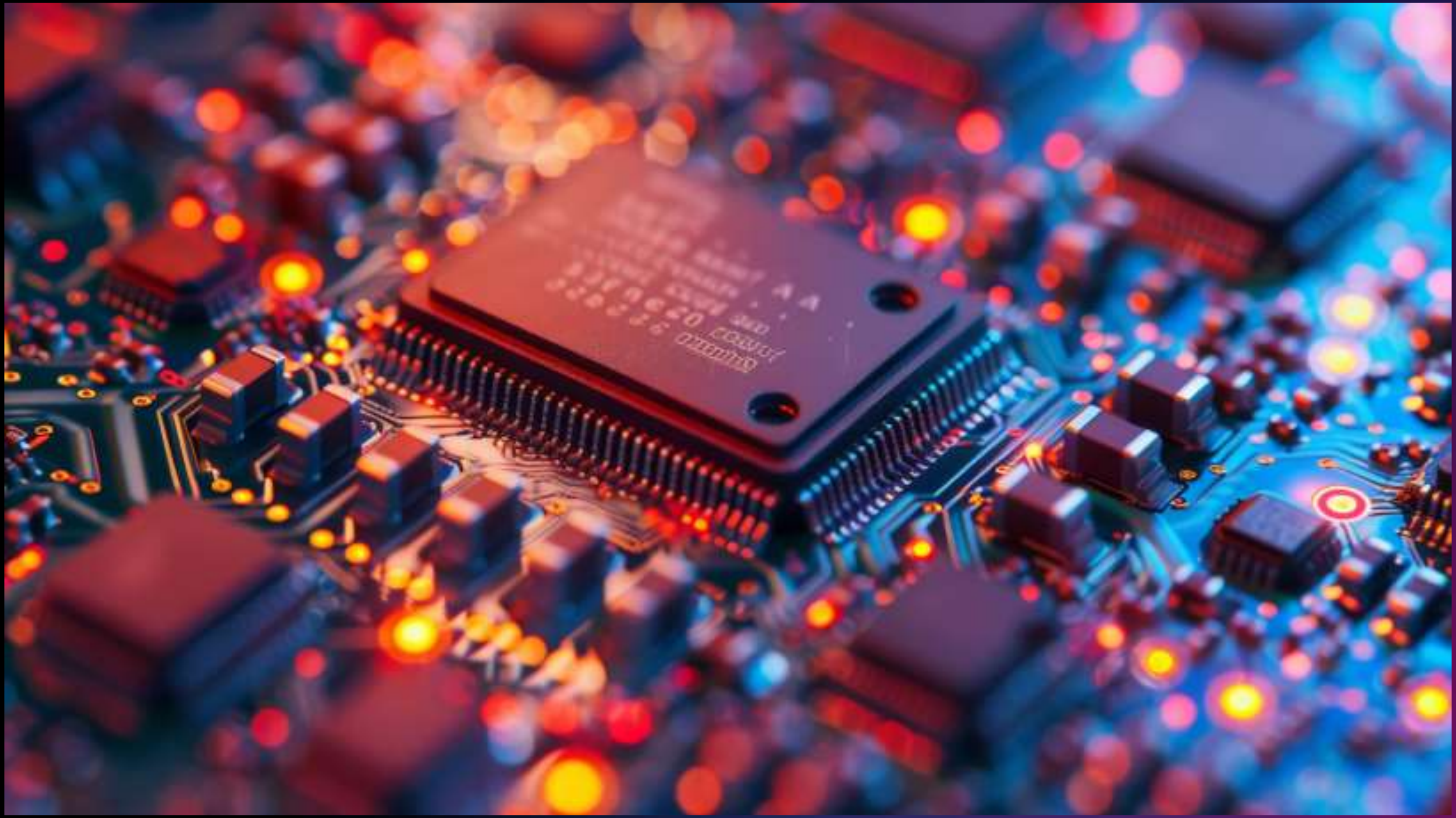
By Layers?

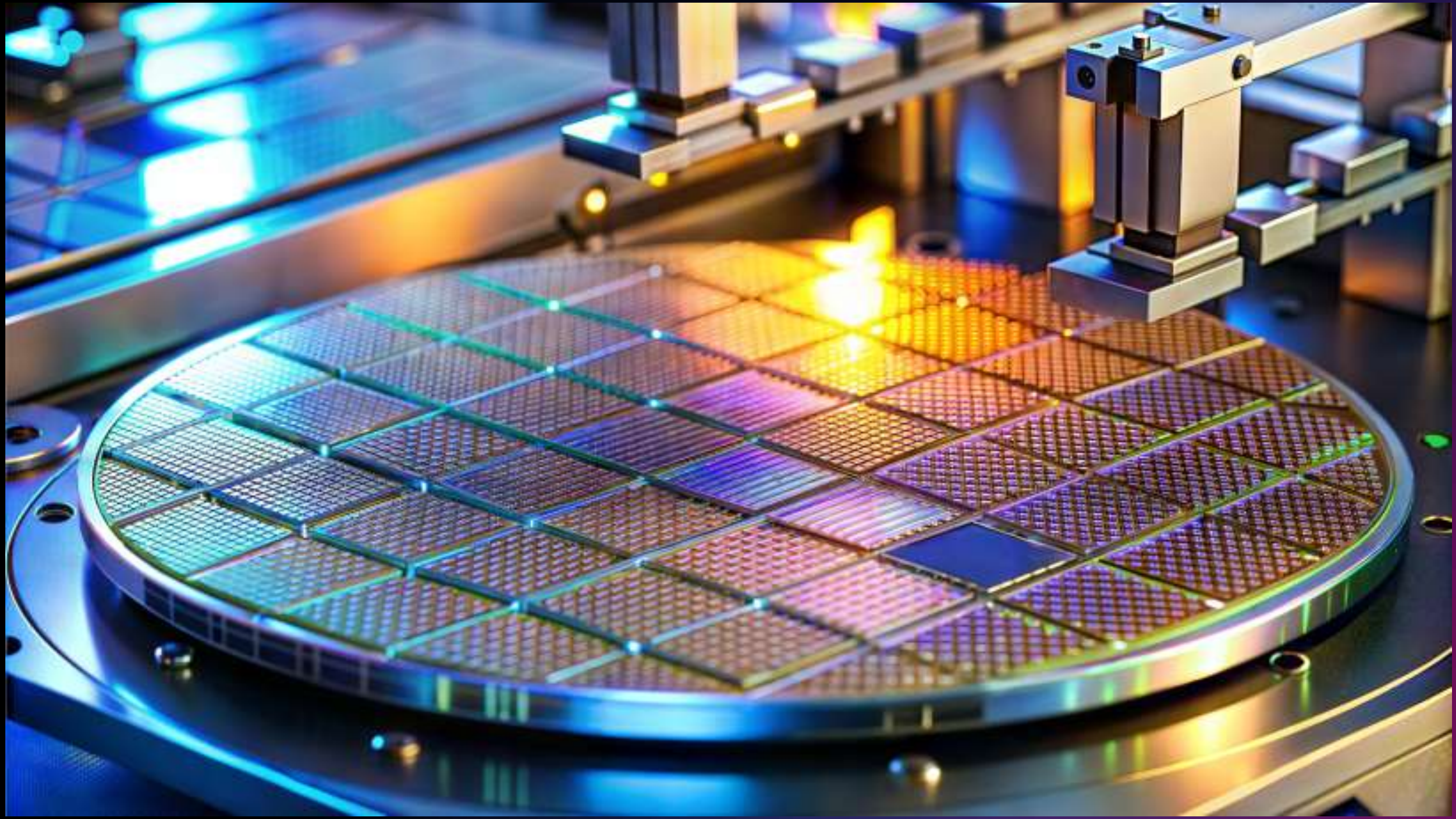
By Design!

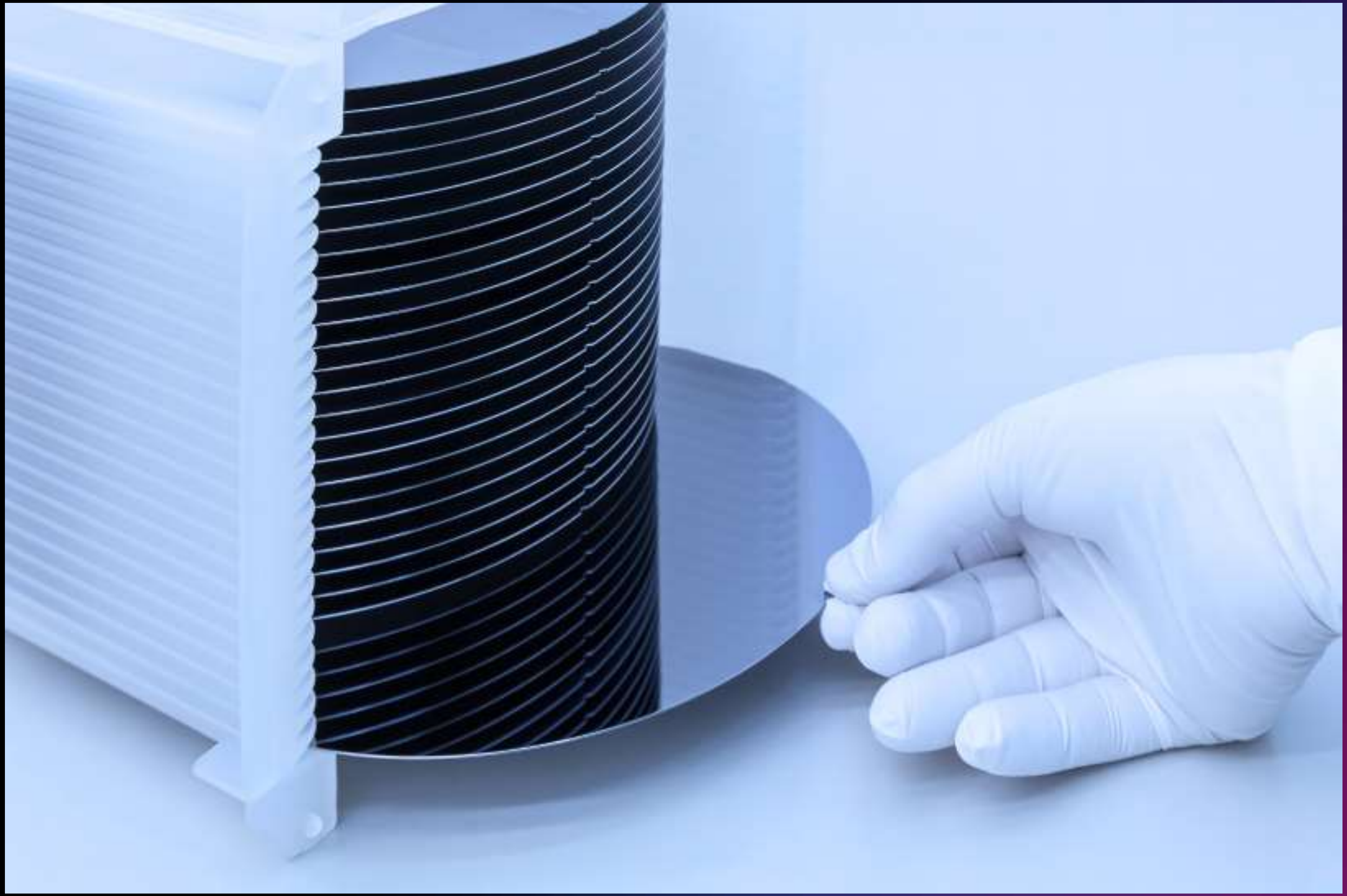
By defense in Depth

<u>With knowledge</u>

Sharing Resources Comes With Security Problems

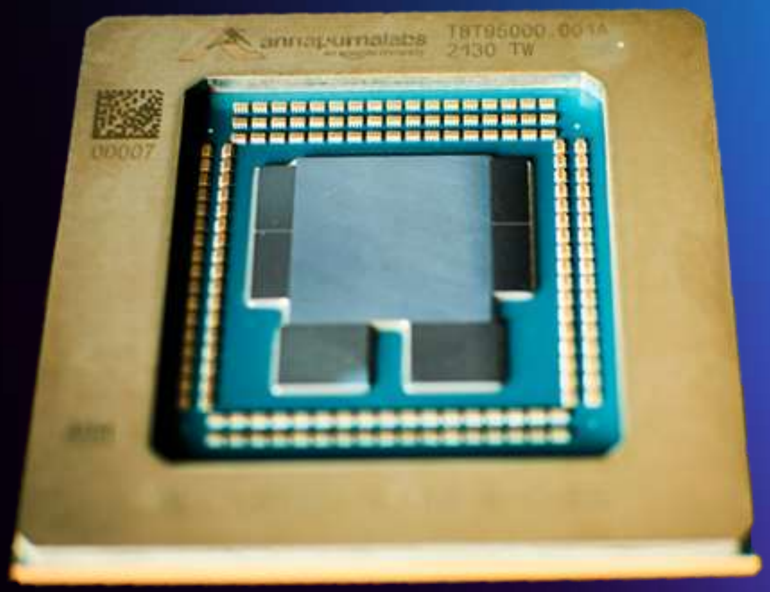






AWS Graviton3

2021



At AWS,
everything starts with

security

SECURE BY DESIGN, AT EVERY LAYER, AT EVERY STAGE

Security **of** Compute

Security **with** Compute

Security **from** Compute

SECURITY OF CONFIDENTIAL COMPUTE

DATA PROTECTION

APPLICATION PROTECTION

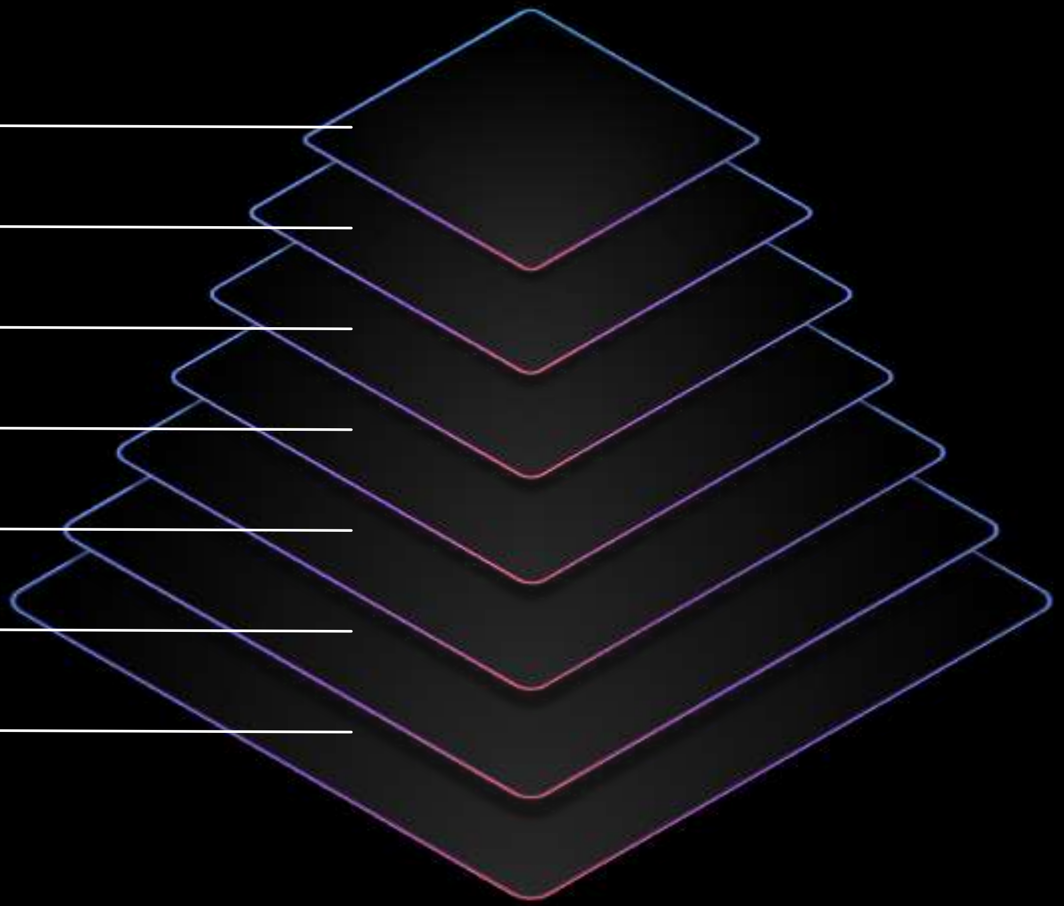
INFRASTRUCTURE PROTECTION

THREAT DETECTION & INCIDENT RESPONSE

IDENTITY & ACCESS MANAGEMENT

NETWORK & EDGE PROTECTION

POLICIES, PROCEDURES, AND AWARENESS



Solved Problems

Sharing Storage

Data at Rest / Storage

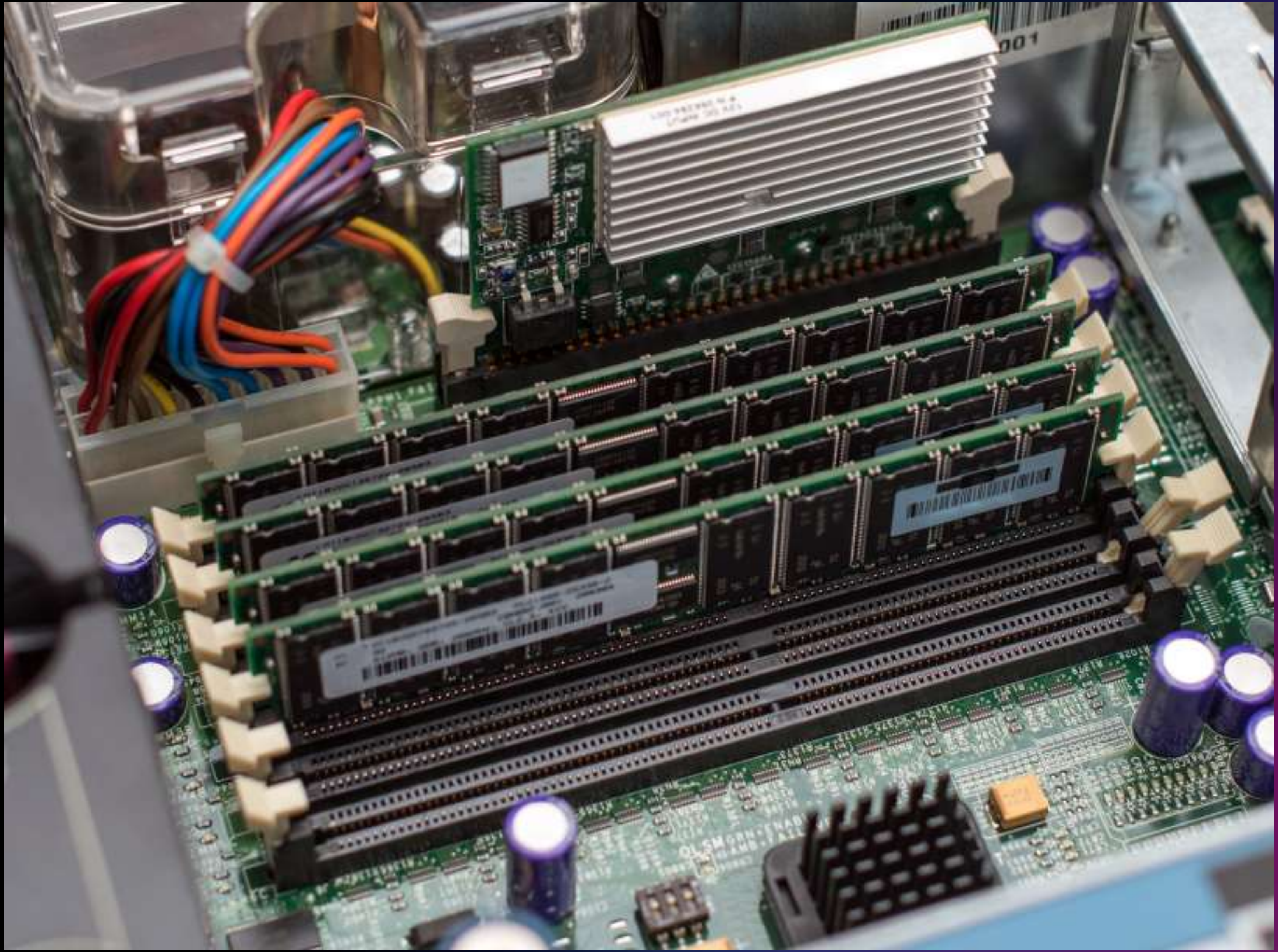
- Ciphertext can’t be seen
- Keys are protected through Envelop Encryption
- Storage is 1:1 Mapped through hardware components
- Ciphertext, Keys, and Algorithms are coming together at -1- place
- Scale
- Speed

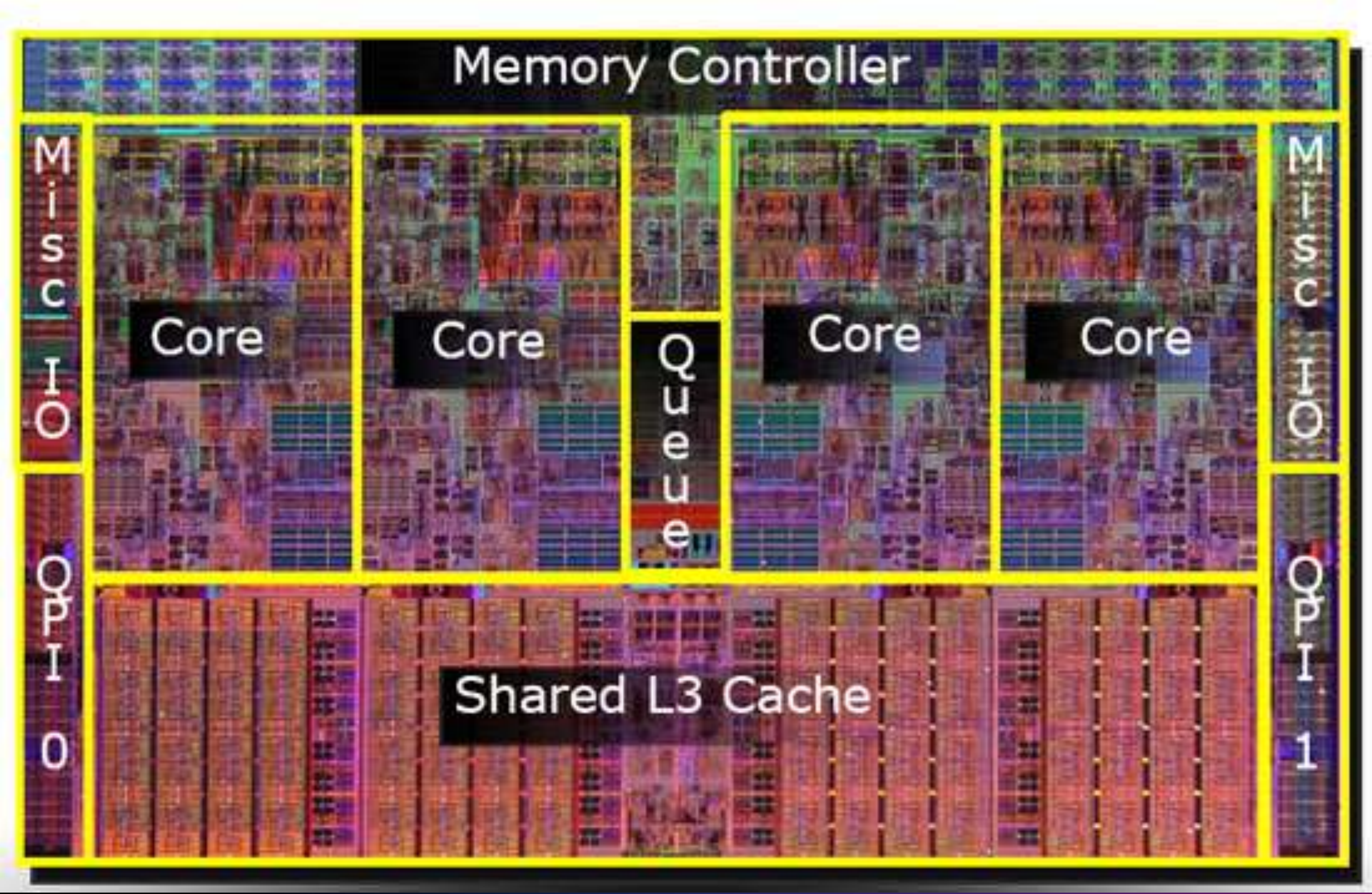
Sharing Network

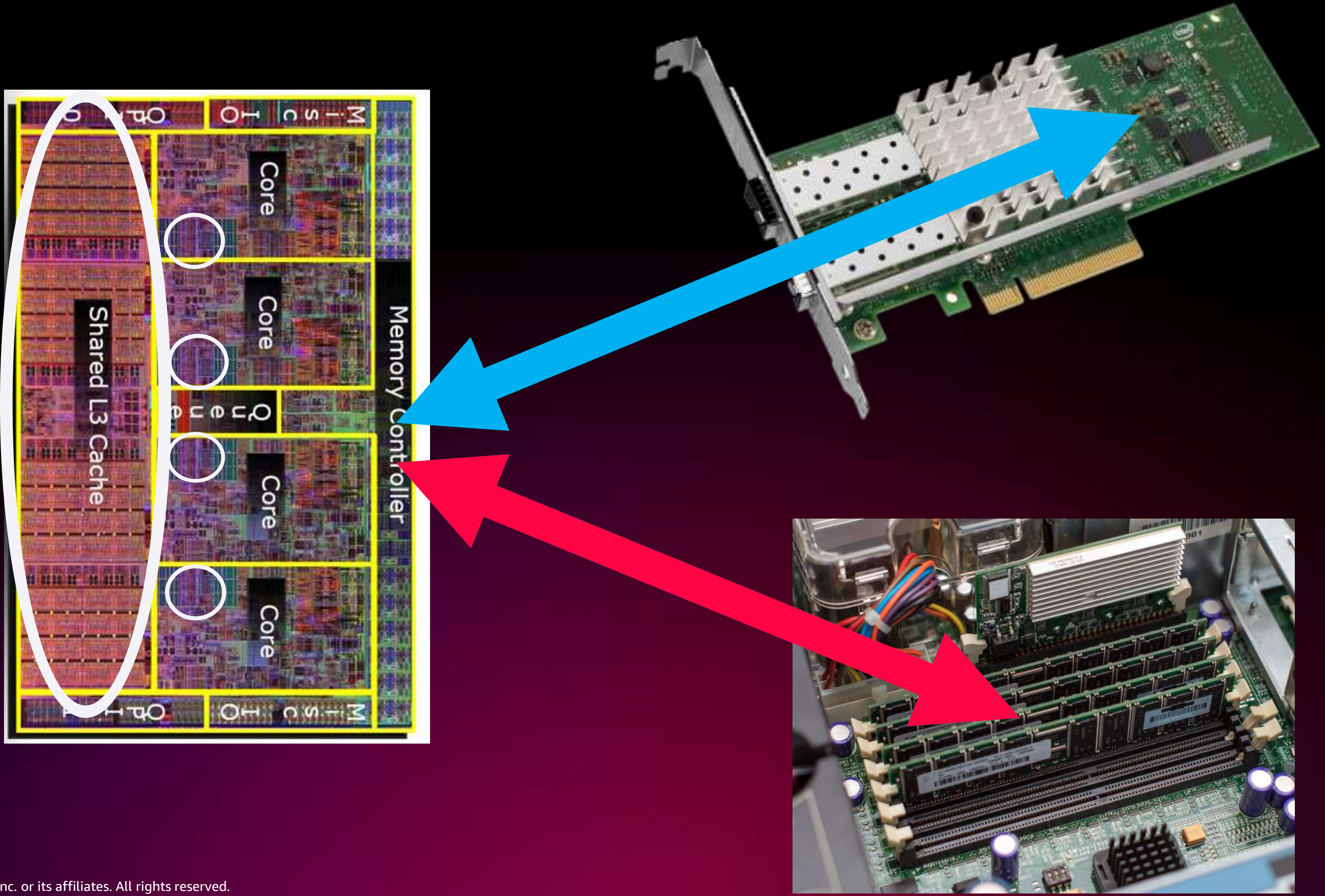
Data in Motion / Networking

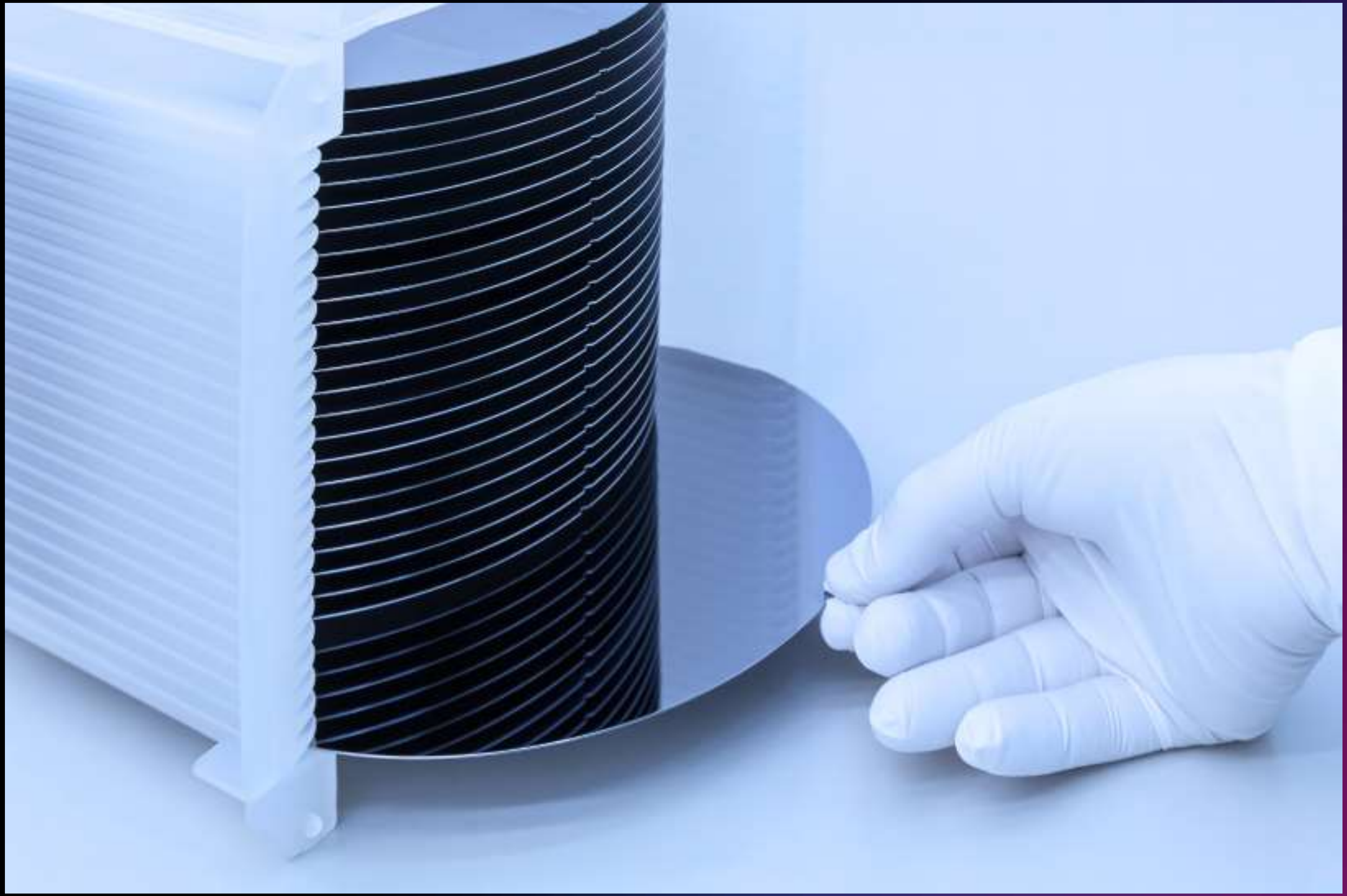
- Transfer creates risks
- Be aware of the “Store now Decrypt later" setup
  - AWS LibLC
  - AWS S2N
- Sender / Receiver Qualification
- Trust chains
- Scale
- Speed

What else?




Memory Controller
Misc IO
Core
Core
Queue
Core
Core
Misc IO
QPI 0
Shared L3 Cache
QPI 1


QPI
IO
Misc
Core
Core
Queue
Core
Core
Memory Controller
Shared L3 Cache
QPI
IO
Misc

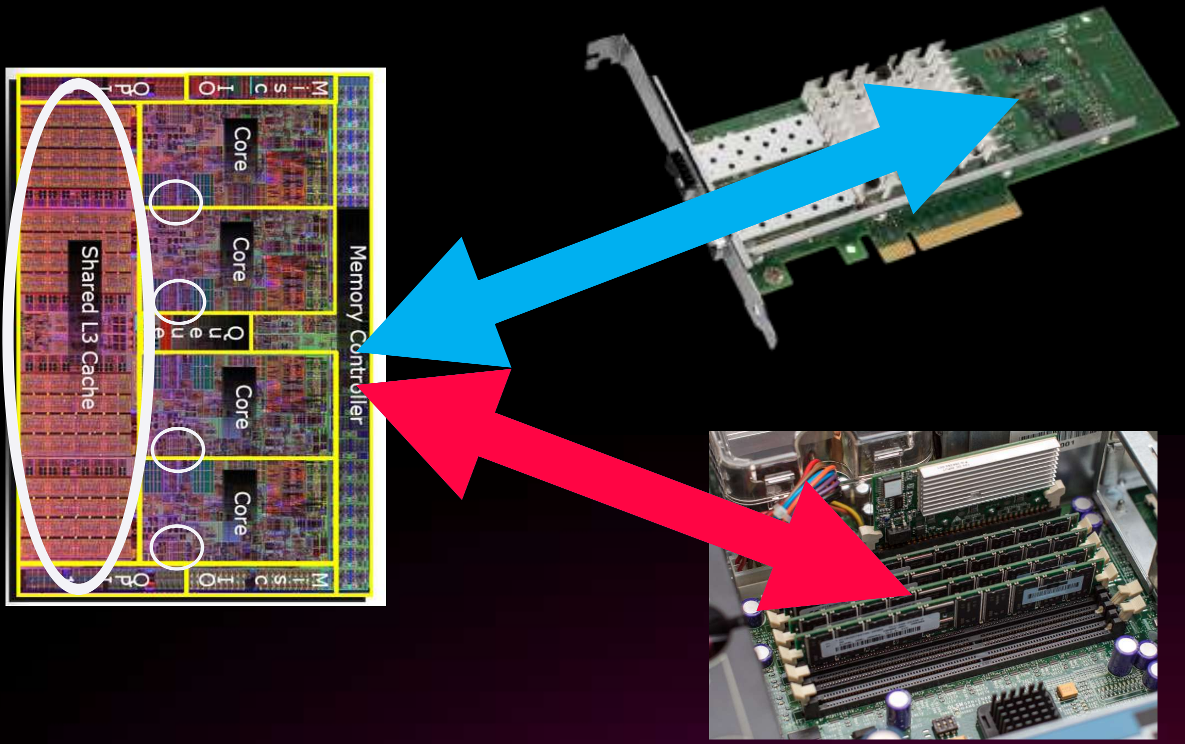


Which components are responsible?

The OS Kernel

together with

the Memory Management Unit of the CPU (MMU)



If things in confidential compute go wrong, they go wrong in the kernel or CPU

- Rowhammer
- Meltdown
- Spectre (TLBs)
- RunC
- cgroups/namespaces

- **Avoid sharing the same kernel with different severities**
- If your “share” partner brings code, have him be on his own CPU core/L1 Cache

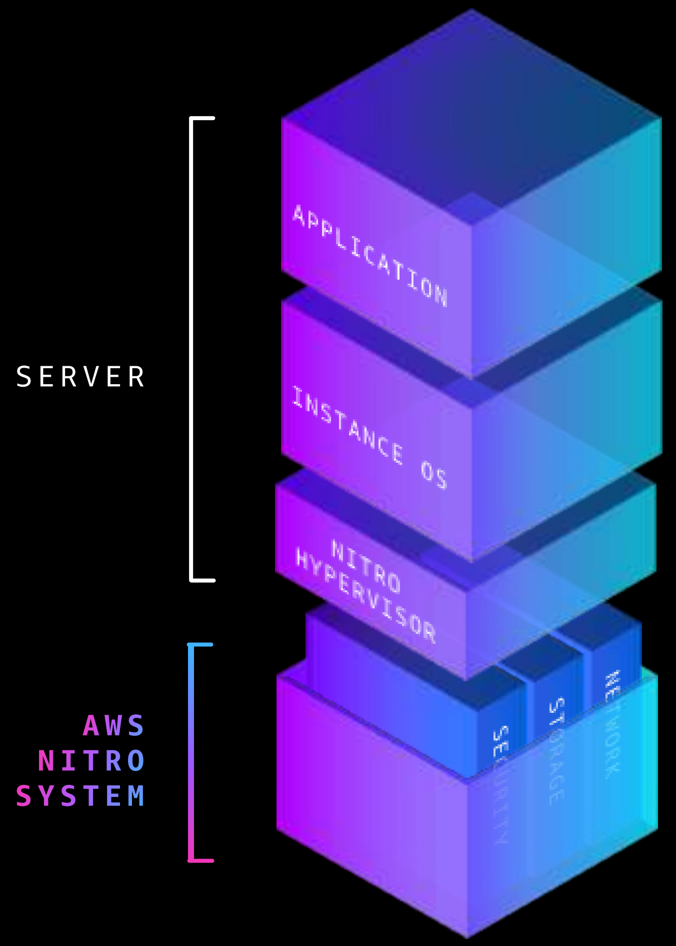
  - Sort

  - Pack

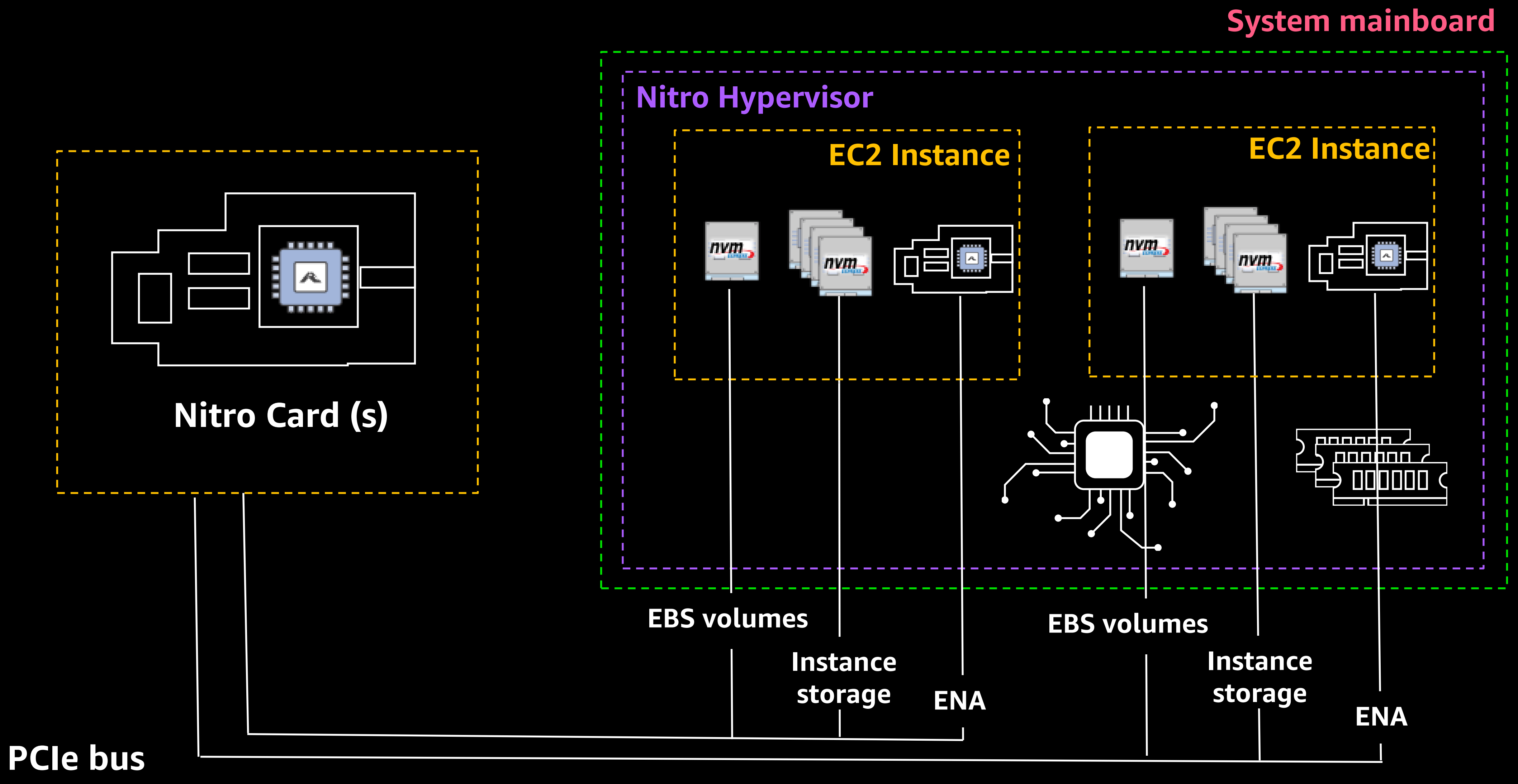
  - Isolate

- Examples on AWS:
  - ***Customers bringing code are not on the same CPU at runtime***
  - AWS Lambda Functions of different customers do not share the same kernel
  - Container Worker nodes (if not Clusters) only for one data type

Confidential Compute requires a **secure foundation**



Nitro virtualized architecture



Transparent hardware encryption

The encryption keys used for EBS, local instance storage, and for VPC networking are only ever present on the system in plaintext within the protected memory of a Nitro Card

Encryption key management

Amazon EBS

- Volumes have independent lifetimes (plus snapshots); therefore, key management via AWS KMS (FIPS 140-2 Level 3 validated HSMs)

Instance storage

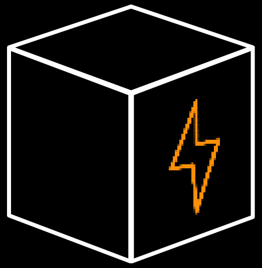
- Locally generated, used, deleted (instance lifecycle)

VPC

- Seed materials regionally generated
- Seeds distributed, not actual secrets; key derived on Nitro device; rotated frequently

Nitro Hypervisor

KVM-based hypervisor with custom memory management and minimized user space



Only executes on behalf of instance, quiescent

With Nitro, the hypervisor is minimal and performant

Nitro Hypervisor: Minimalization

The Nitro Hypervisor has been deliberately minimization

Intentionally excludes

- Networking stack
- General purpose file system implementations
- Peripheral driver support
- SSH server
- Shell
- Etc.

Nitro Hypervisor: Primary functions

1. Receive virtual machine management commands (start, stop, and so on) sent from the Nitro Controller
2. Partition memory and CPU resources by utilizing hardware virtualization features of the server processor
3. Assign SR-IOV virtual functions provided by Nitro hardware interfaces (NVMe block storage for EBS and instance storage, Elastic Network Adapter [ENA] for network, and so on) through PCIe to the appropriate VM

The Security Design of the AWS Nitro System

Publication date: **November 18, 2022** (*Document revisions* (p. 27))

Abstract

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. The AWS Nitro System is the underlying platform for all modern EC2 instances. This whitepaper provides a detailed description of the security design of the Nitro System to assist you in evaluating EC2 for your sensitive workloads.

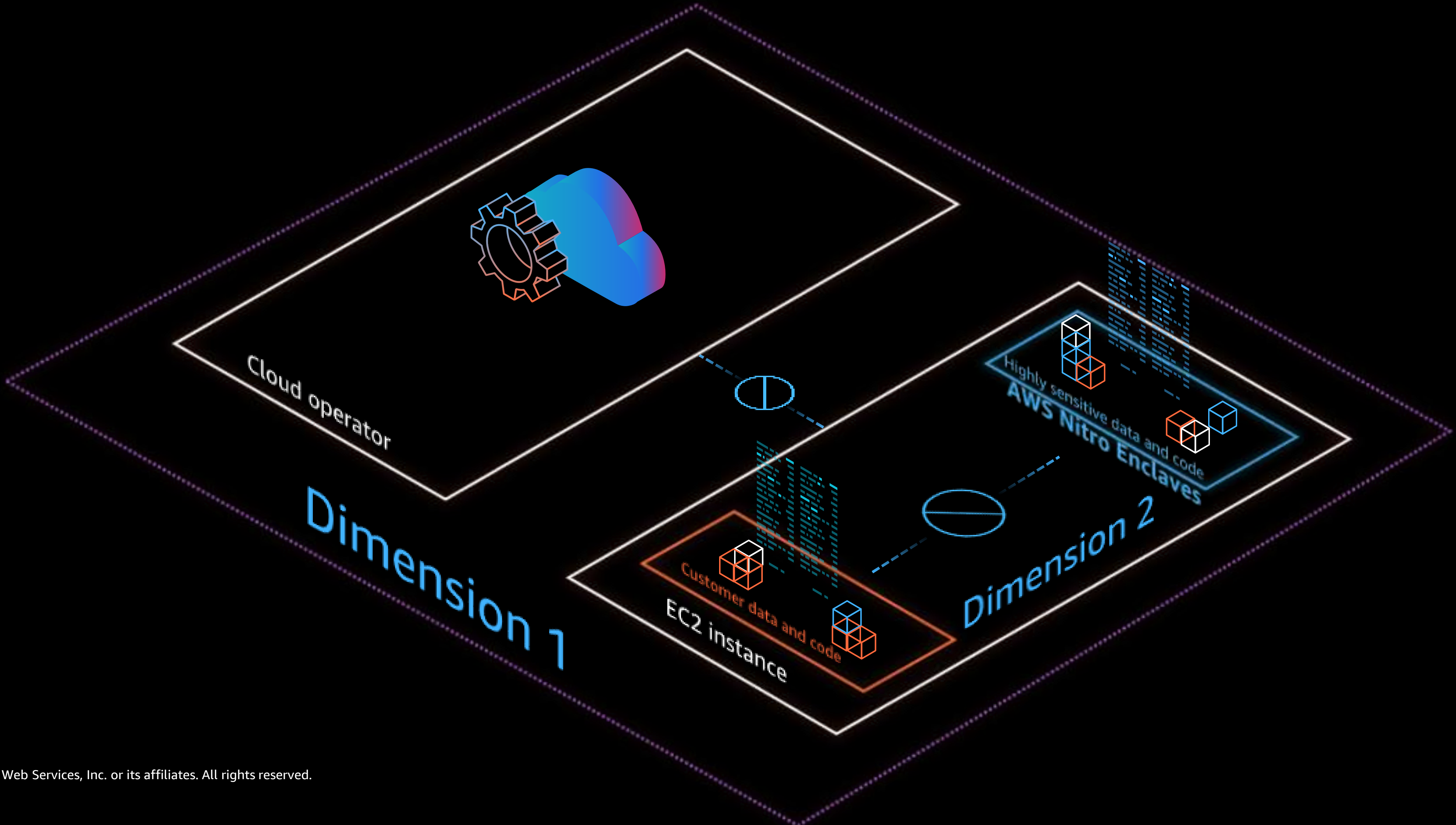
https://a.co/hYWhsH9

- Detailed review of the security design of the three primary components of the AWS Nitro System
- Deep dive on the AWS Nitro System integrity protections, tenant isolation model, and no-operator access design

Geometry -> Dimensions

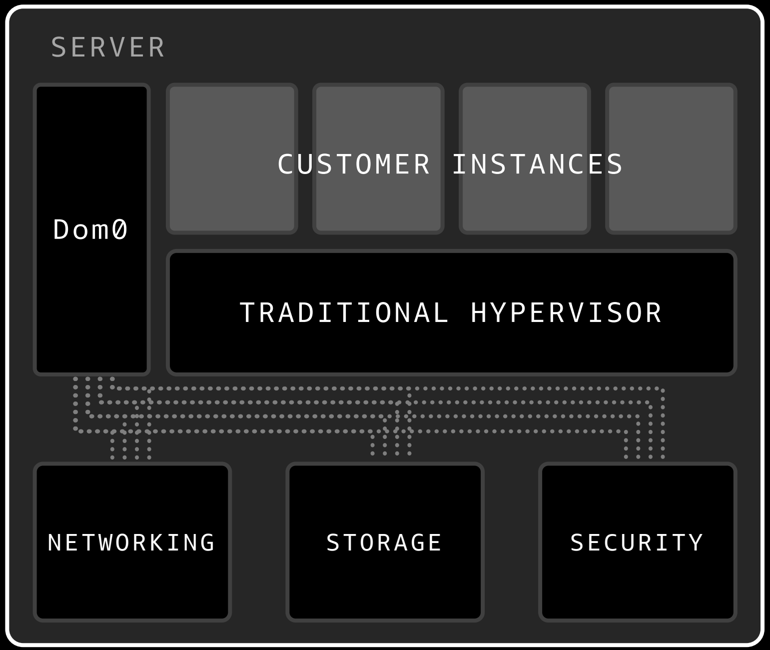
What is confidential computing?

PROTECTING CUSTOMER CODE AND SENSITIVE DATA IN USE

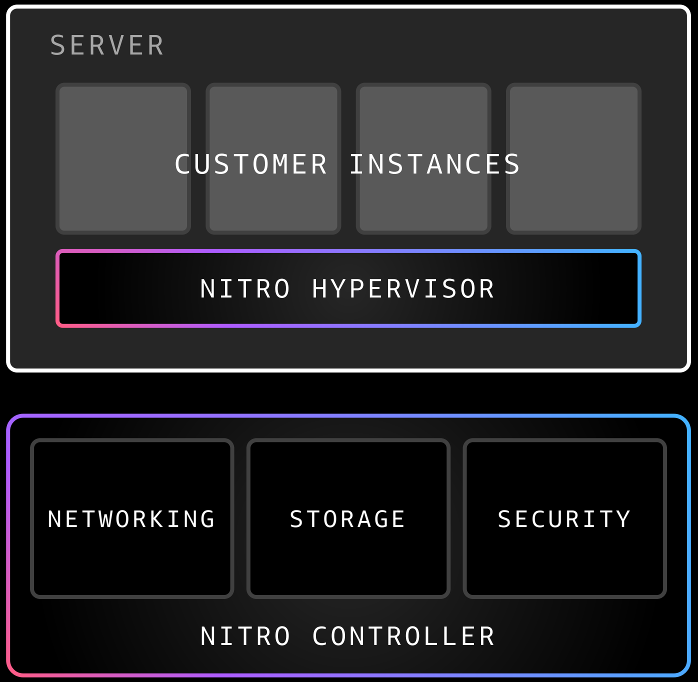


Confidential Compute requires **verifiable control**

STANDARD CLOUD VIRTUALIZATION



AWS NITRO SYSTEM



No operator access (no Dom0, no SSH)

Hardware-based isolation

System boot attestation

Passive communication design

AWS Nitro Enclaves

CREATE ISOLATED COMPUTE ENVIRONMENTS TO FURTHER PROTECT AND PROCESS HIGHLY SENSITIVE DATA

FULLY ISOLATED

Enclaves are hardened and constrained environments that can only be accessed through secure local channels

SECURE

Attestation is integrated with AWS Key Management Service (AWS KMS) to ensure sensitive data is only accessed by the Nitro enclave

USE CASES

Further protect and process highly sensitive data, such as personally identifiable information (PII), healthcare, and financial data

Confidential Compute requires **core protection**

Incident response

AWS Security Hub

Amazon GuardDuty

Threat detection

Data protection

AWS KMS

AWS Shield Advanced

Network & app protection

Thank you!

Bertram Dorn