Ceviche: Capability-Enhanced Secure Virtualization of Caches

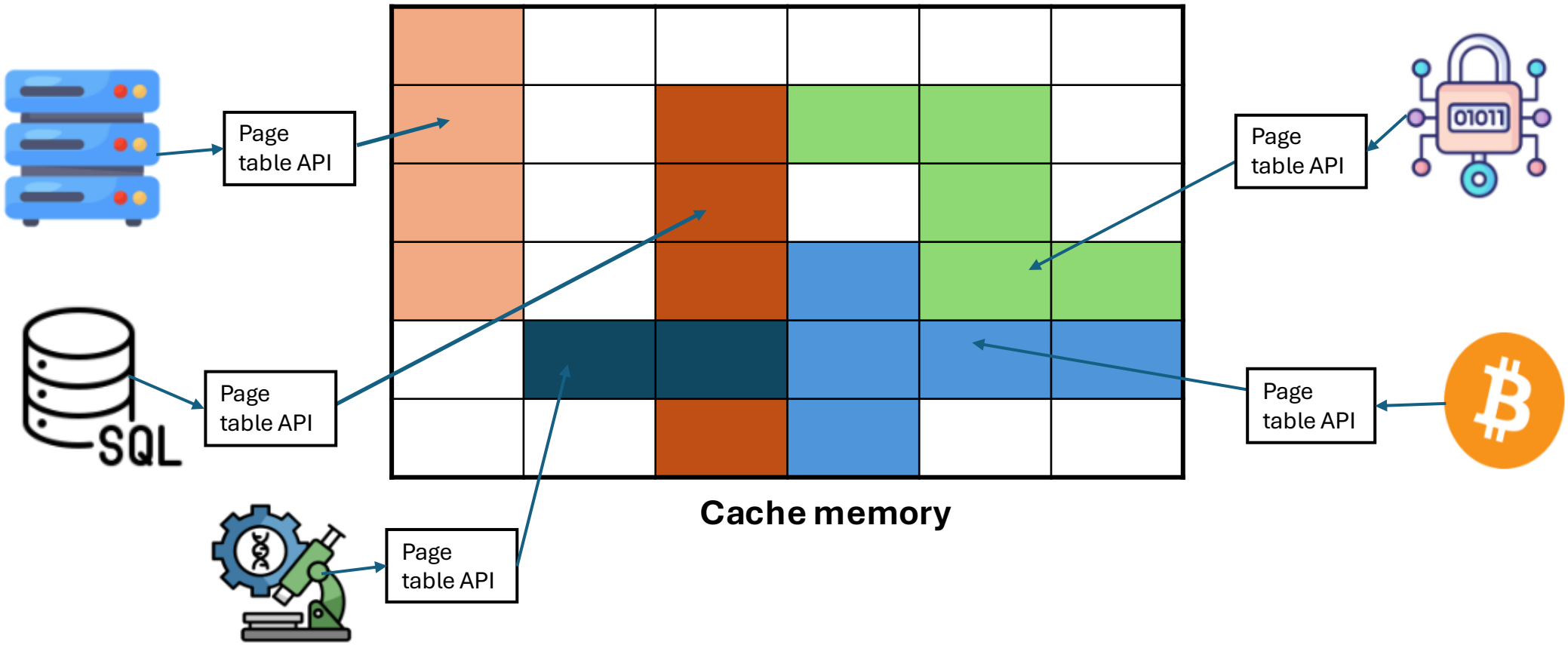
Arnabjyoti Kalita, University of Virginia, mje6jj@virginia.edu

Yilong Yang, University of Virginia, yyilong@virginia.edu

Alenkruth Krishnan Murali, University of Virginia, alenkruth@virginia.edu

Ashish Venkat, University of Virginia, venkat@virginia.edu

Virtualization



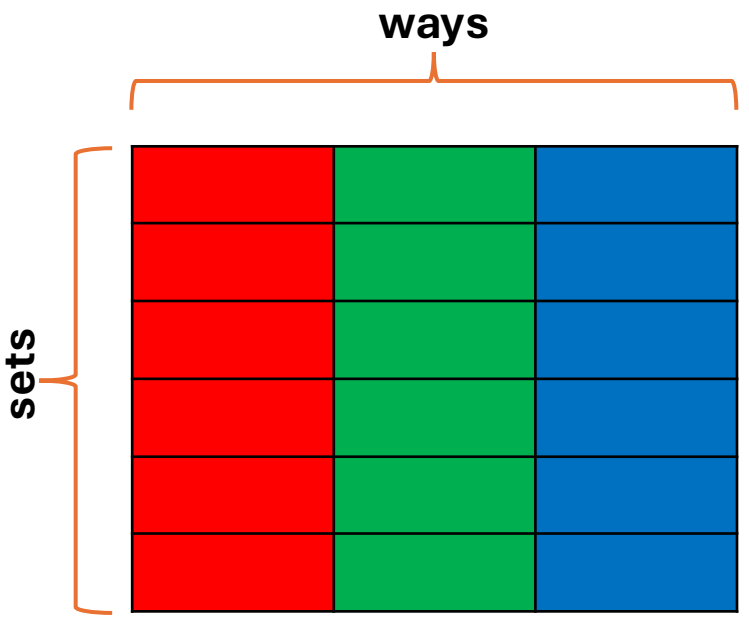
Cache virtualization allows flexible allocation and efficient utilization of cache resources.

But…what about security?

Cache virtualization enables efficient sharing of the cache.

Sharing cache lines ➡ interference ➡ information leakage

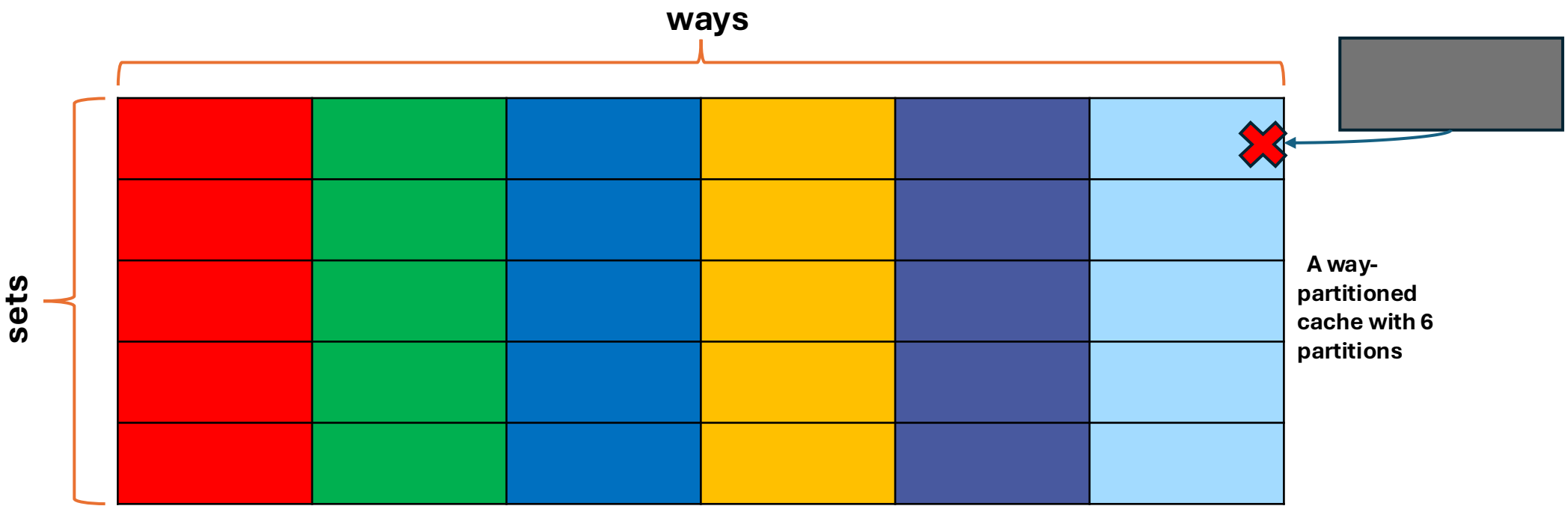
Solution: partition cache resources by cache sets or cache ways.



Cache way-partitioning

Problems with existing secure caches

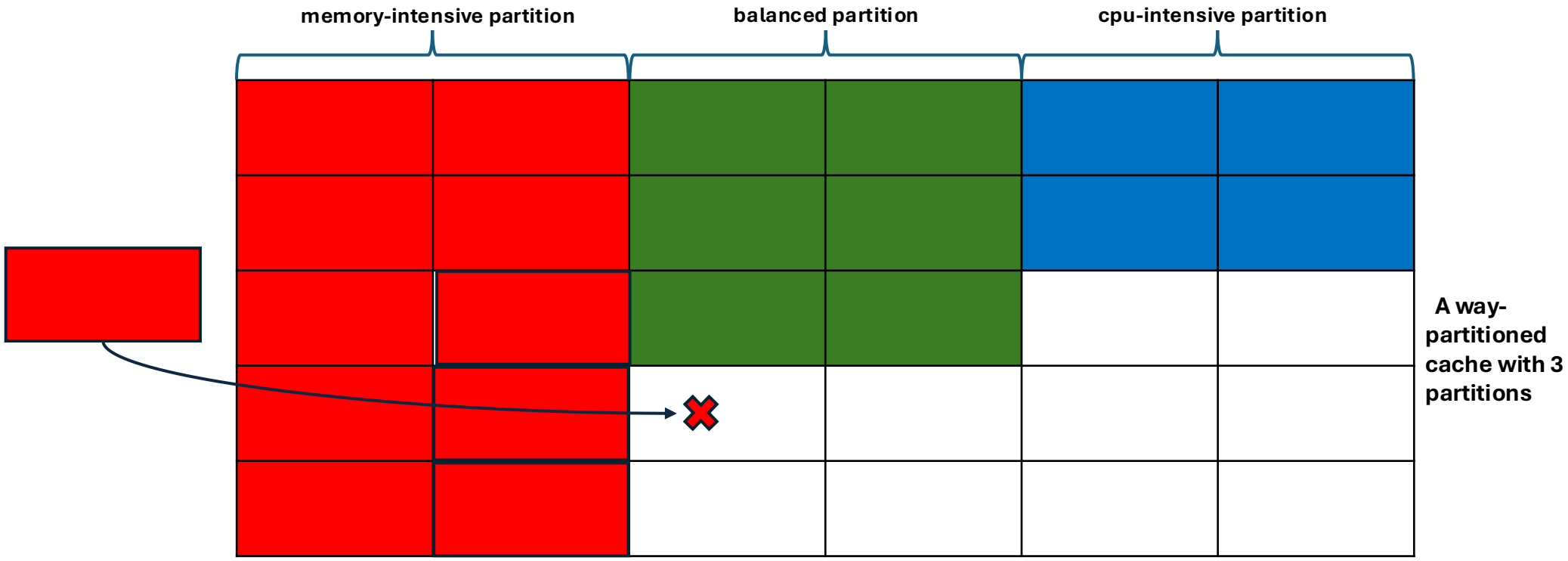
Existing partitioned caches are not scalable.



Number of cache partitions is limited by the number of ways or number of sets in the cache.

Problems with existing secure caches

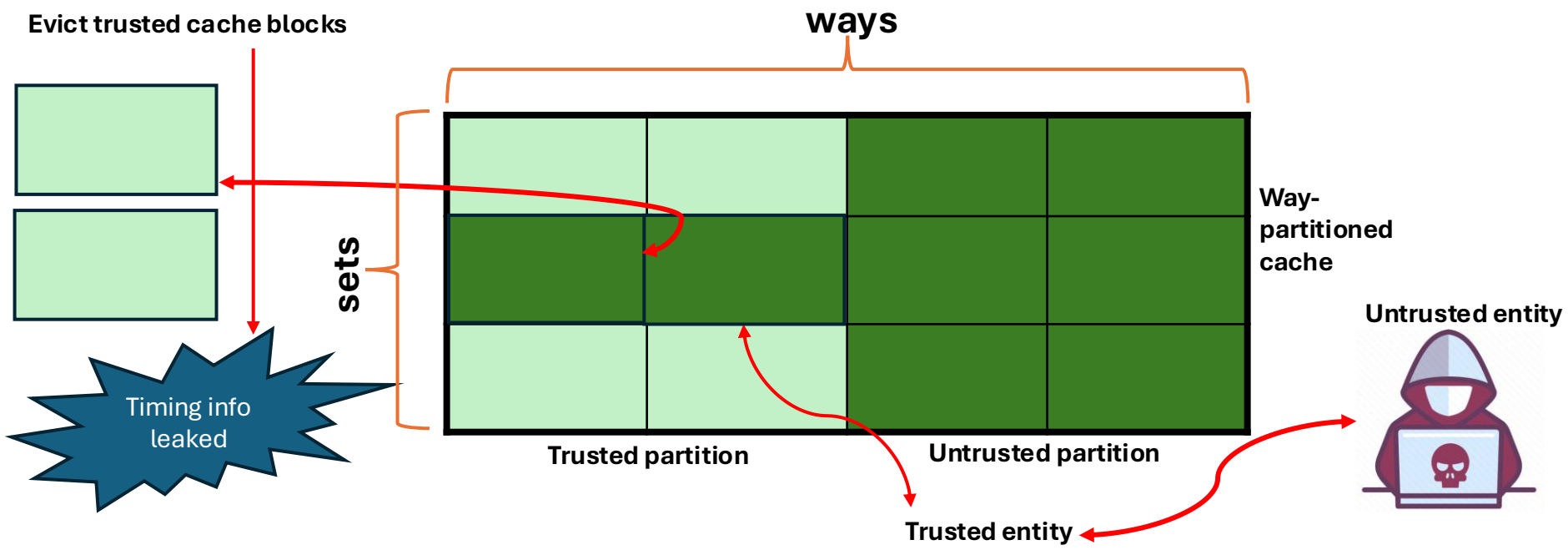
Existing partitioned caches lead to low utilization.



The variability of cache demand in applications will lead to poor utilization of partitions in a cache.

Problems with existing secure caches

Existing partitioned caches cannot protect against indirect attacks like confused-deputy.



The attacker deputizes trusted system entities to bypass existing isolation strategies in the cache.

Capabilities?

Capability = object name + user's permitted privileges.

Capabilities are unforgeable until they are explicitly revoked.

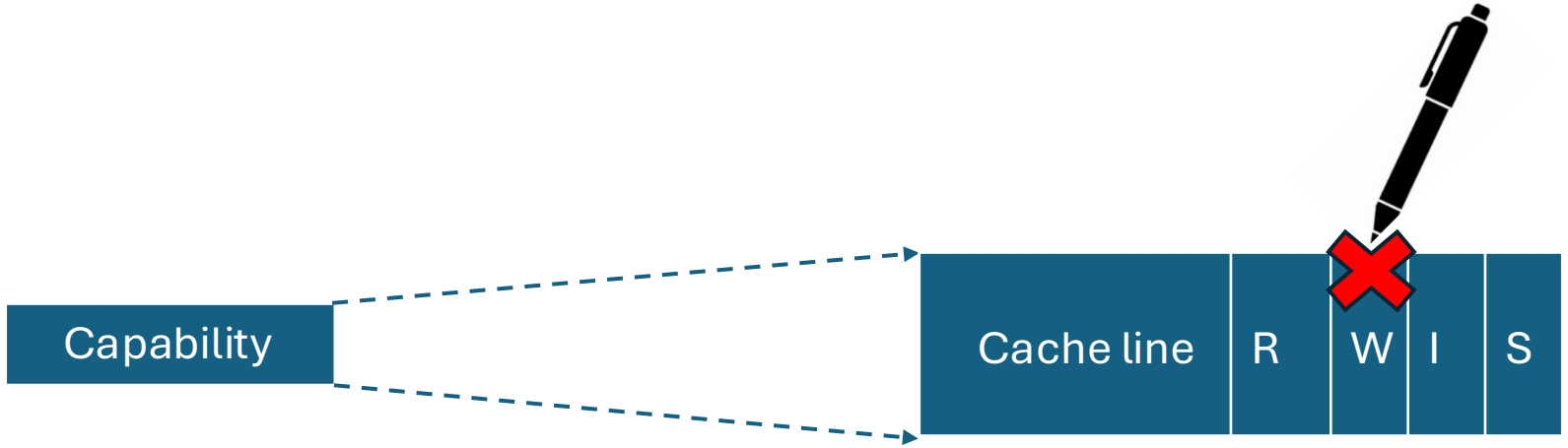
Capabilities can also be shared between users through mutual agreement.

Ceviche: A capability-based solution

Least-Privilege Access: The capability enforces a least privileged access to an authorized object, i.e. cache line.

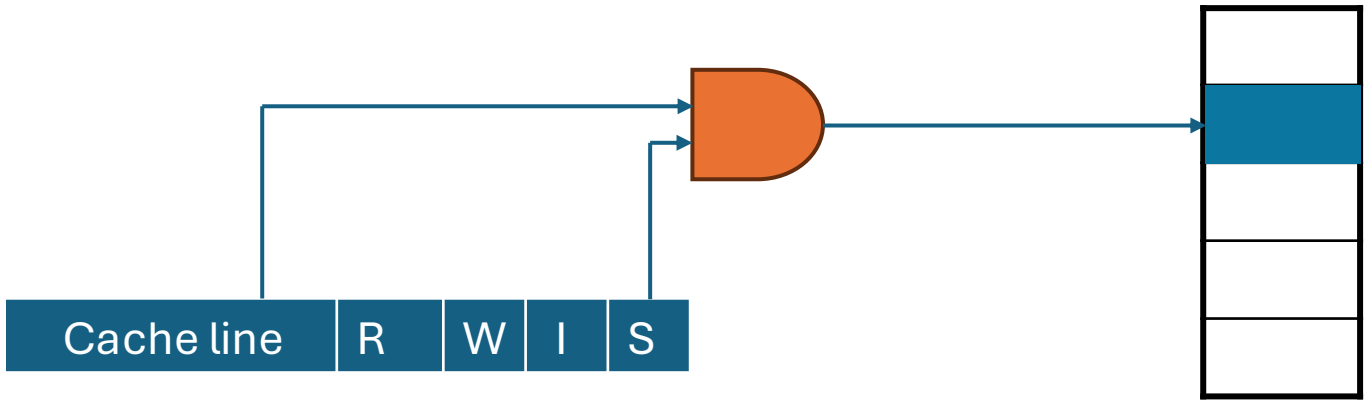
Ceviche: A capability-based solution

Unforgeability: Access rights baked into a capability cannot be changed once it is burned.



Ceviche: A capability-based solution

Intentionality: All cache operations need to be explicitly enforced/permitted by the capability.

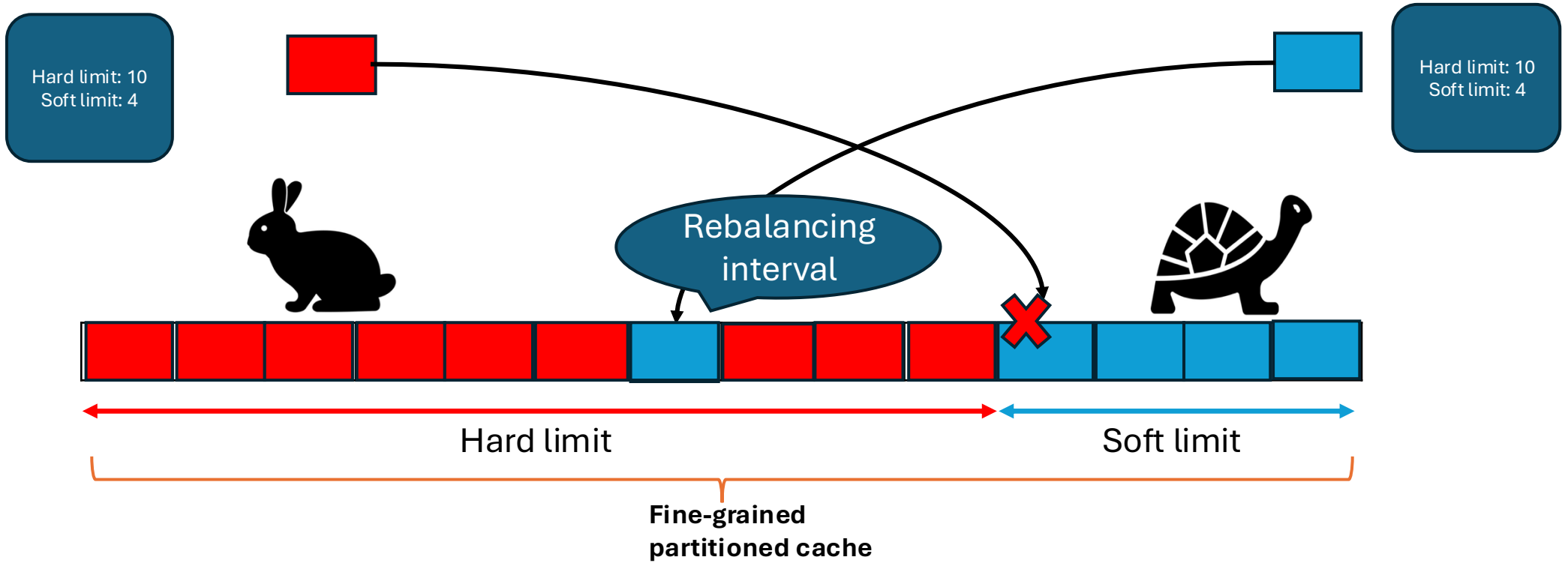


Ceviche: A capability-based solution

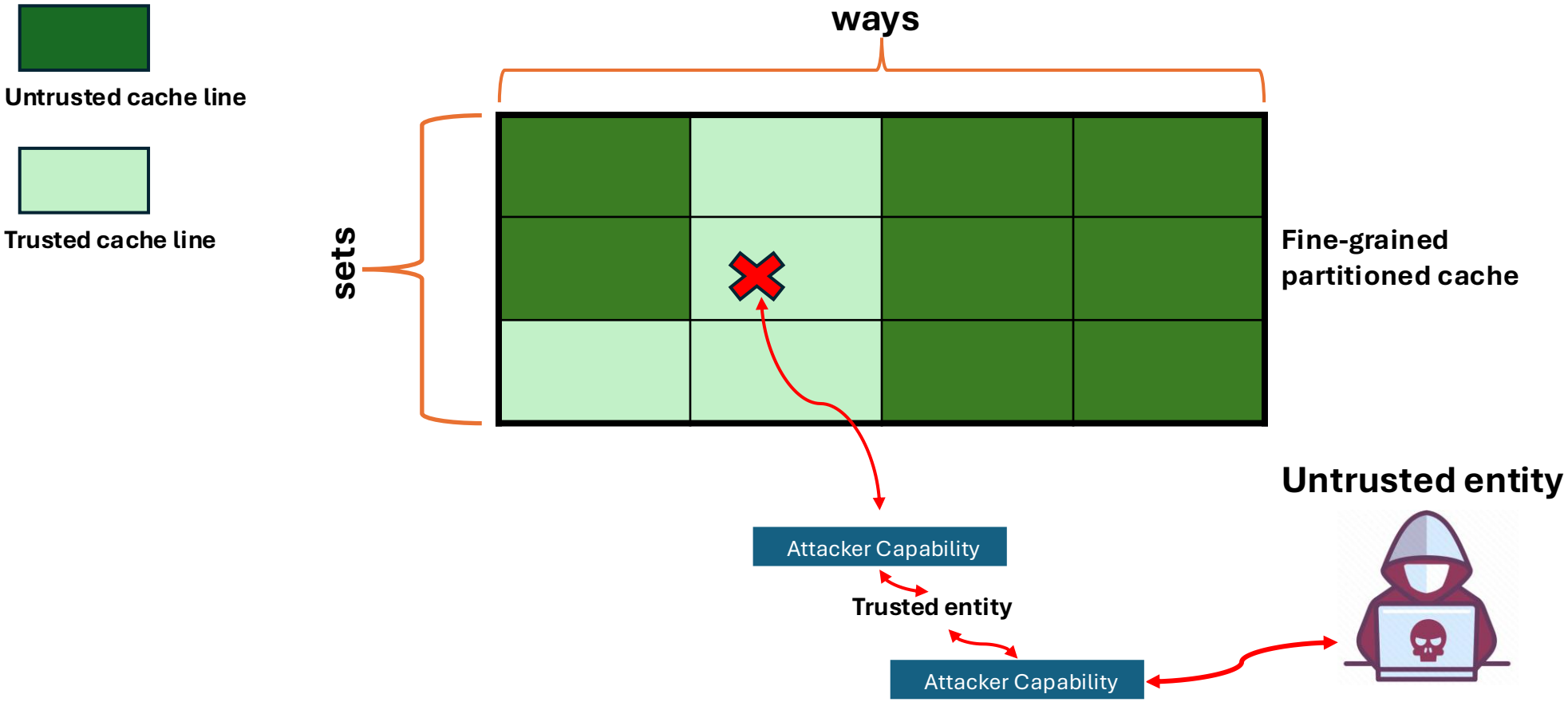
Address Independence: Capabilities break the tight coupling between the virtual/physical address and the actual physical location of the cache line in the cache.

Ceviche: A capability-based solution

Fairness and Availability: Fairness and availability is ensured through the enforcement of hard and soft limits.

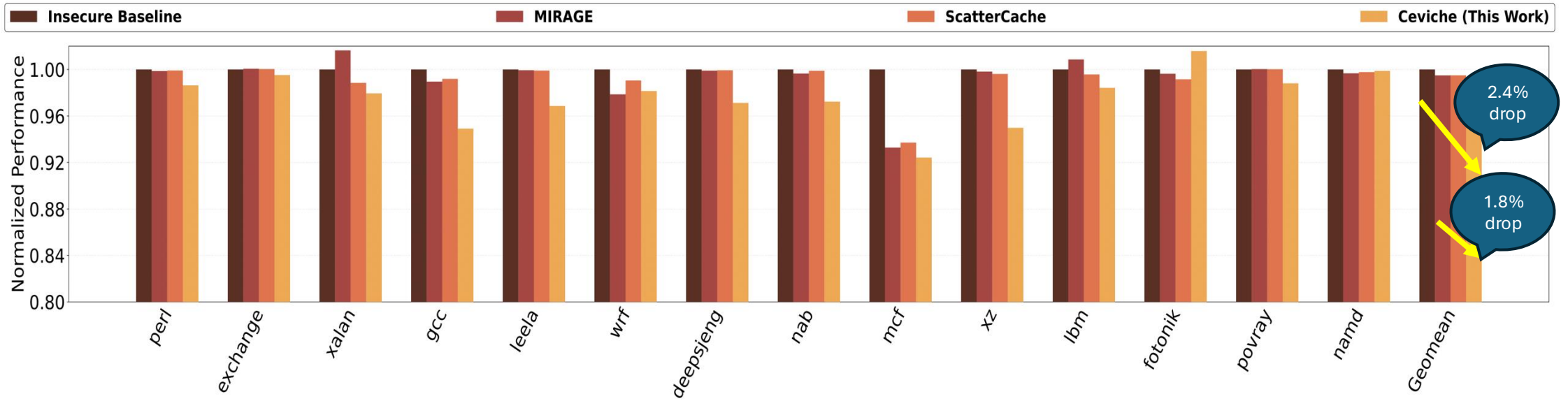


Ceviche: Preventing Confused-Deputy



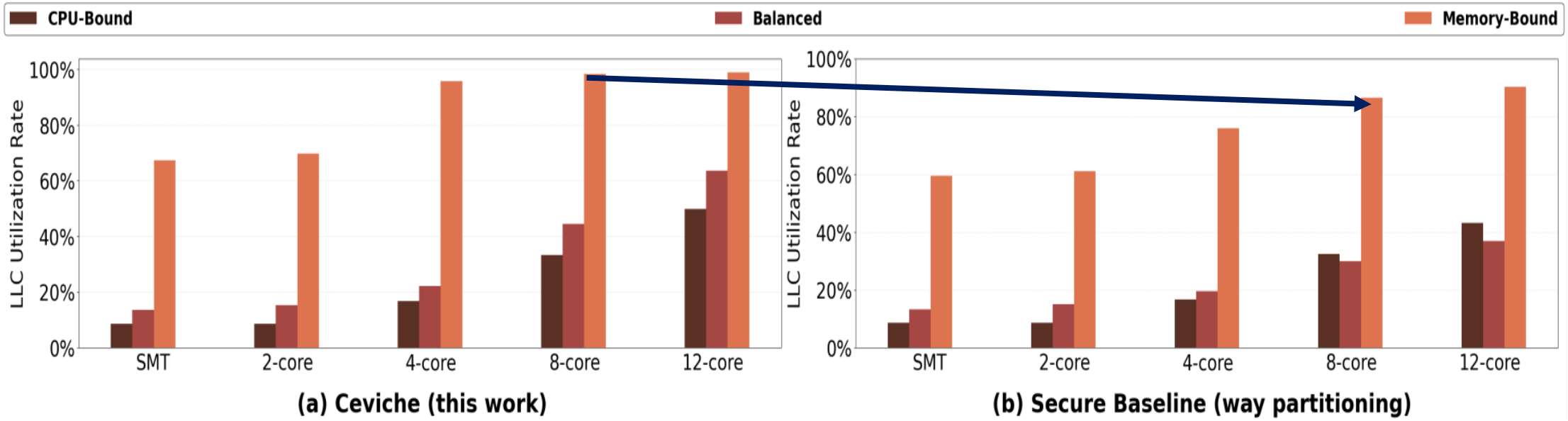
The trusted entity intentionally uses the attacker's capability, which prevents bypassing existing security

Results: Single-Threaded Performance



Ceviche incurs a 1.8% performance drop against state-of-the-art caches while providing protection across all cache levels and a wider variety of attacks.

Results: Multi-Threaded Utilization



With higher number of cores (4, 8, 12), a statically partitioned cache has a lower utilization of the LLC.

Conclusion

Ceviche is a novel secure virtualization solution that protects against a wide variety of attacks.

Ceviche ensures conflict-averse fully-associative allocation and fast direct-mapped lookup.

Ceviche can gracefully scale to multiple domains and larger caches, at modest performance degradation.

All icons used from : flaticon.com and nounproject.com