HOW IS CAP CONSISTENCY DIFFERENT FROM ACID CONSISTENCY?

ACID consistency is all about database rules. If a schema declares that a value must be unique, then a consistent system will enforce uniqueness of that value across all operations. If a foreign key implies deleting one row will delete related rows, then a consistent system will ensure the state can’t contain related rows once the base row is deleted.

CAP consistency promises that every replica of the same logical value, spread across nodes in a distributed system, has the same exact value at all times. Note that this is a logical guarantee, rather than a physical one. Due to the speed of light, it may take some non-zero time to replicate values across a cluster. The cluster can still present a logical view by preventing clients from viewing different values at different nodes.

The most interesting confluence of these concepts occurs when systems offer more than a simple key-value store. When systems offer some or all of the ACID properties across a cluster, CAP consistency becomes more involved. If a system offers repeatable reads, compare-and-set or full transactions, then to be CAP consistent, it must offer those guarantees at any node. This is why systems that focus on CAP availability over CAP consistency rarely promise these features.