NoSQL & NewSQL

Immanuel Trummer
itrummer@cornell.edu
www.itrummer.org

Reading List

- https://cassandra.apache.org/

- **Data Consistency Properties and the Trade-offs in Commercial Cloud Storages: the Consumers' Perspective**, H. Wada et al, CIDR 2011

- https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed/, E. Brewer.

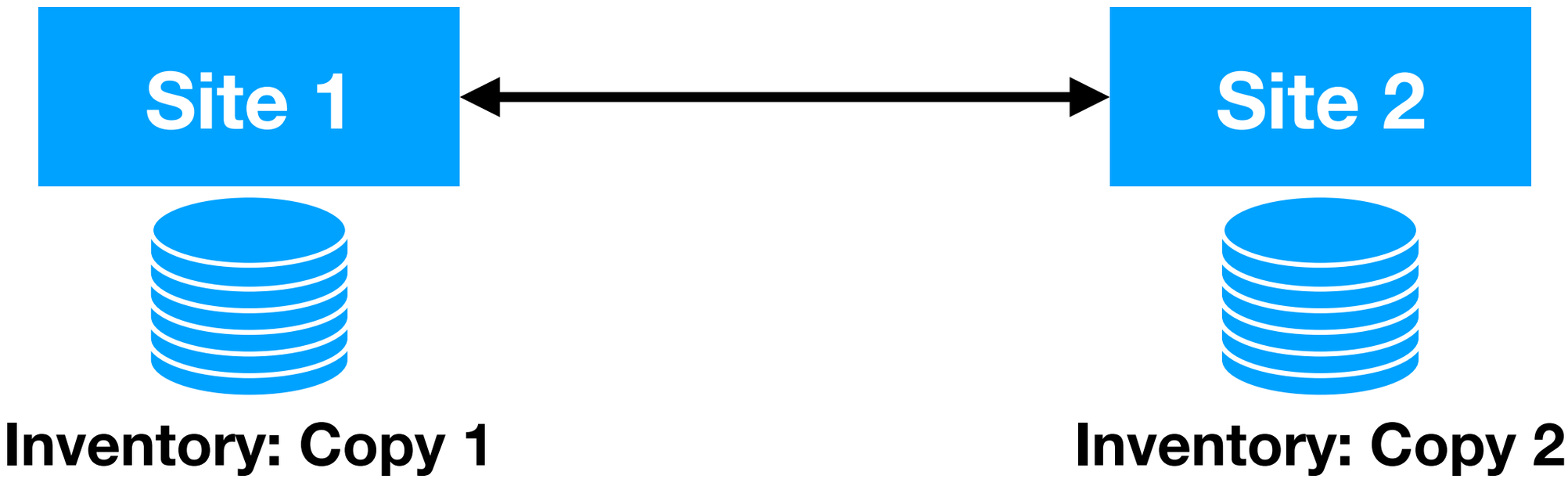
- **H-Store: A High-Performance, Distributed Main Memory Transaction Processing System**, R. Kallman et al., 2008.

- **NewSQL vs. NoSQL for New OLTP**, M. Stonebraker, 2011. https://www.youtube.com/watch?v=uhDM4fcI2aI

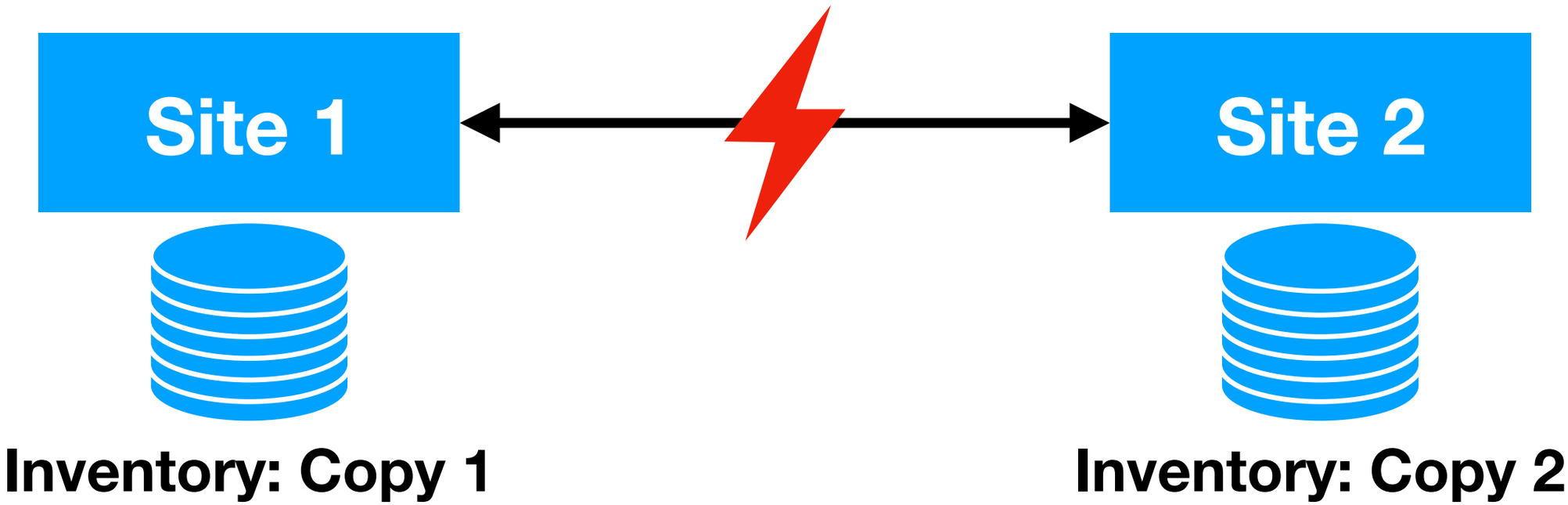
Terminology Warning

- Consistency so far: database satisfies all **constraints**
- Now: consistency means ~ all replicas appear in **sync**
- Terminology from **distributed systems** community

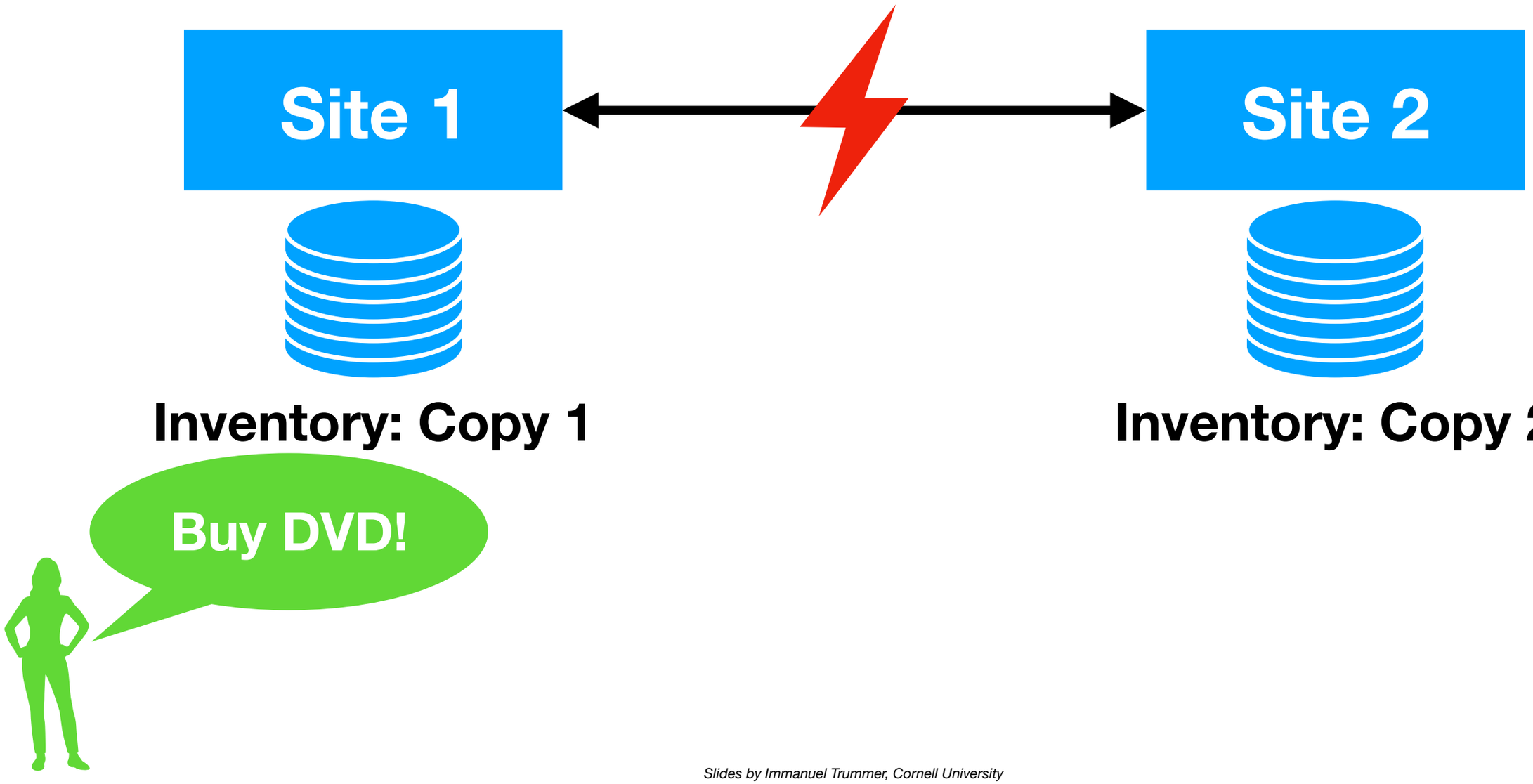
Consistency vs. Availability



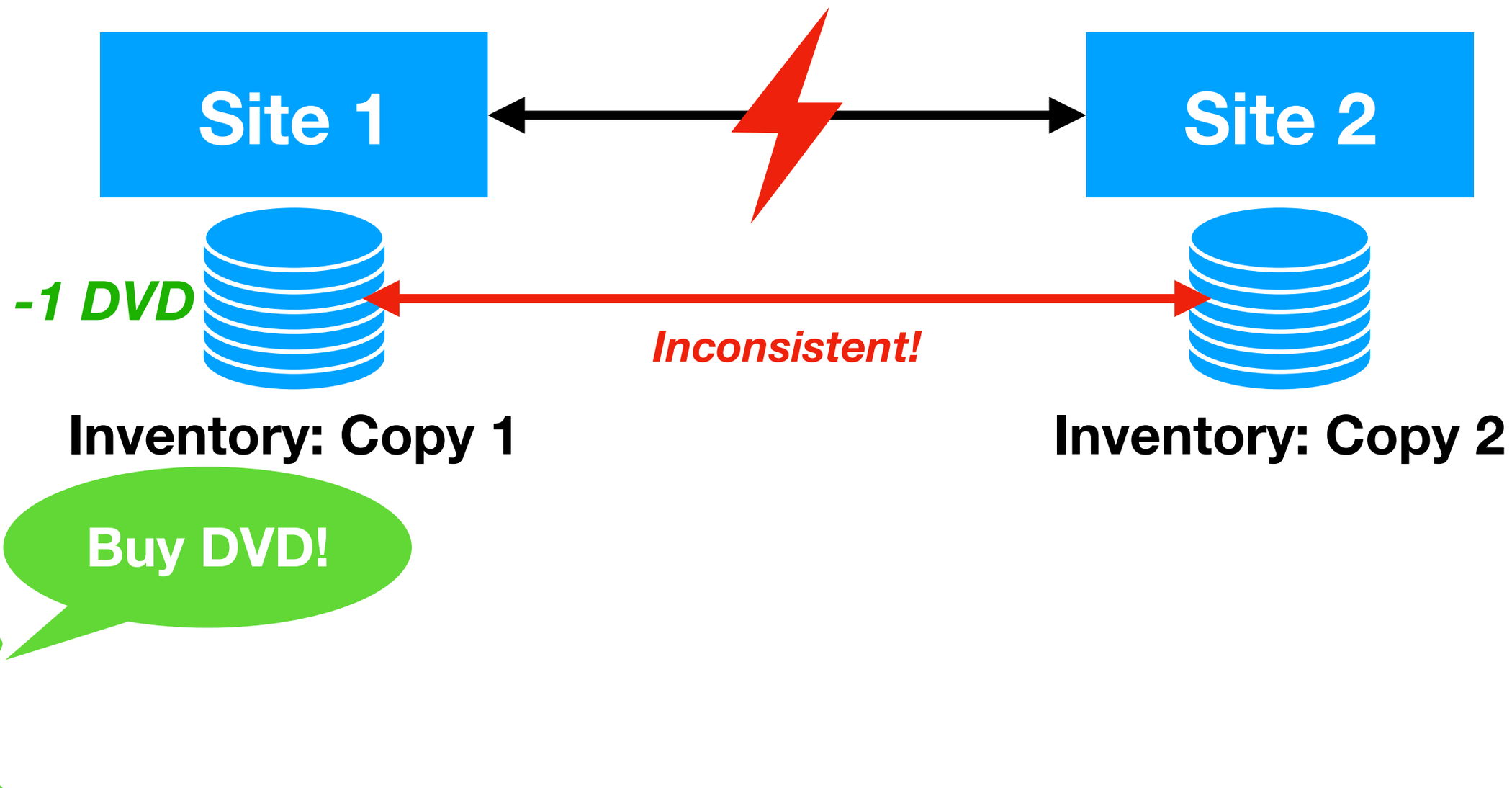
Consistency vs. Availability



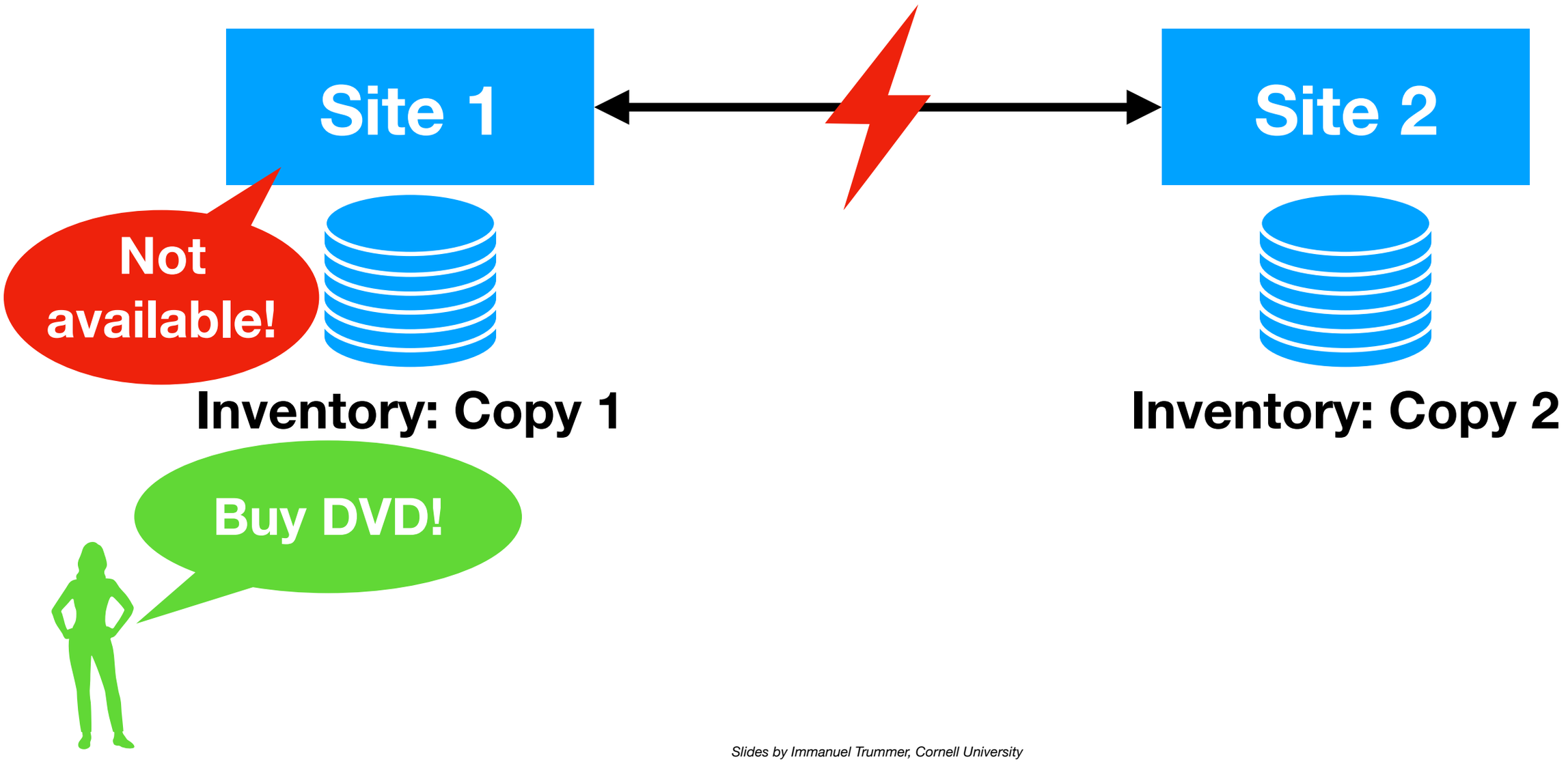
Consistency vs. Availability



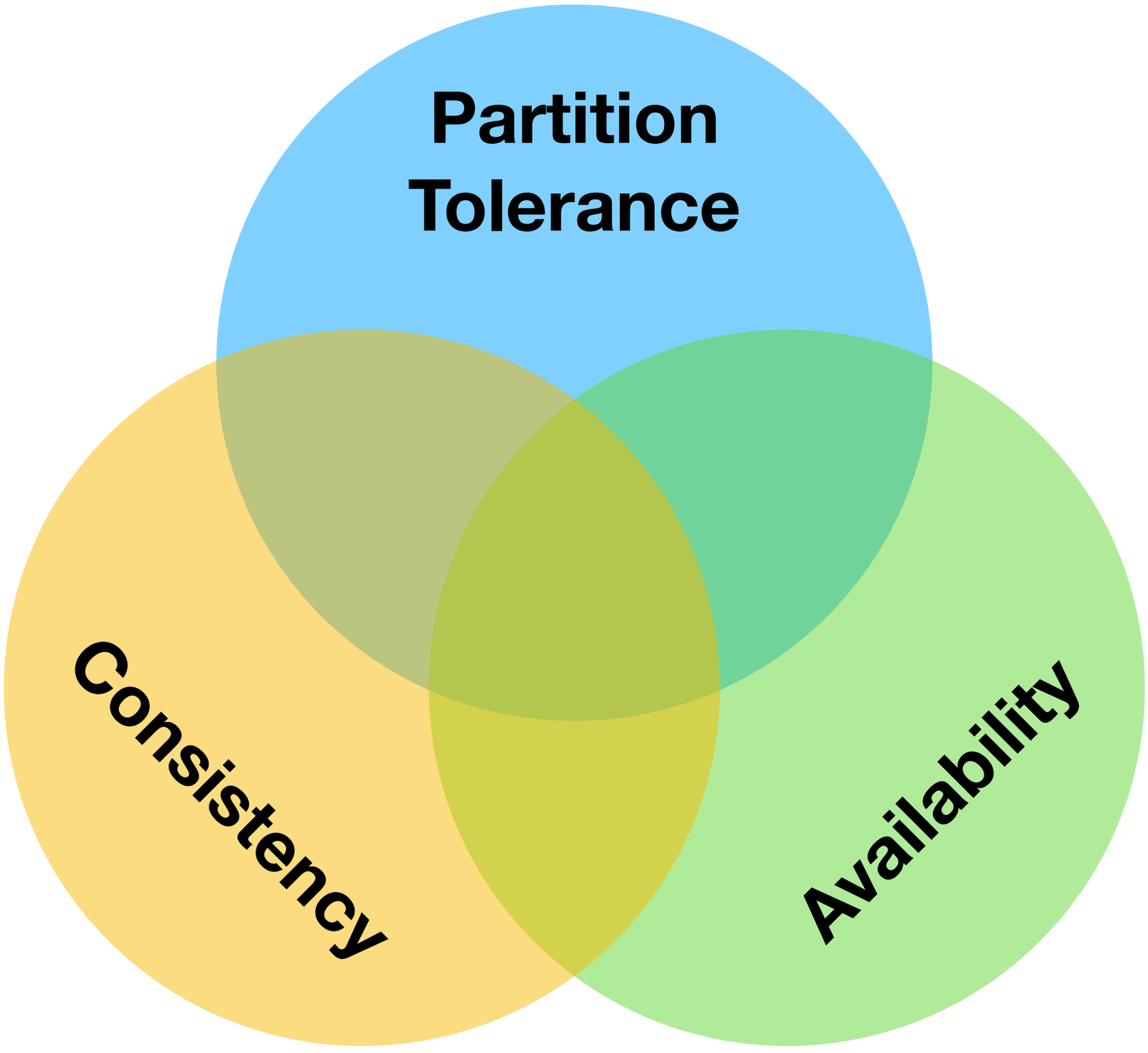
Consistency vs. Availability



Consistency vs. Availability

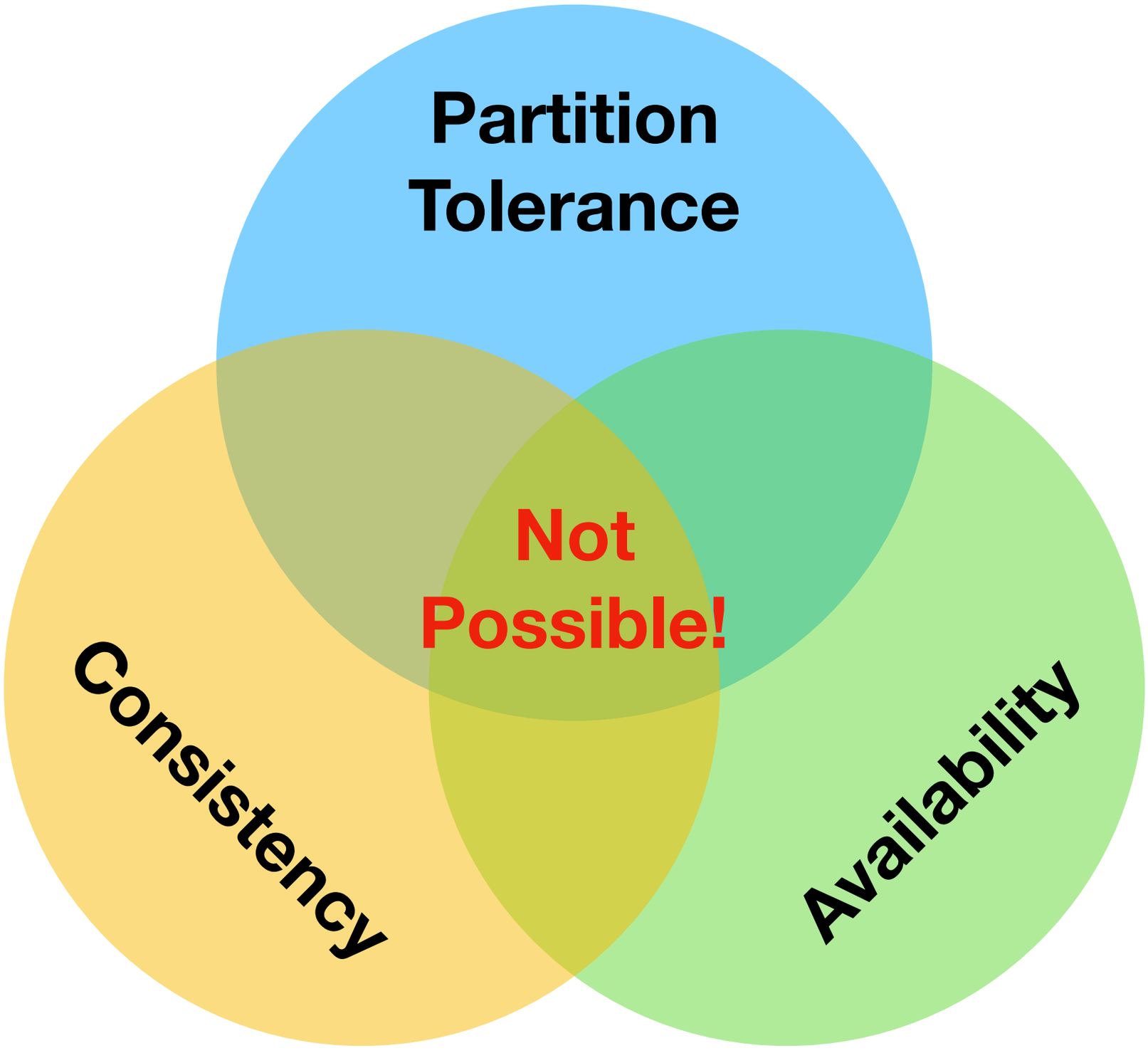


The CAP Theorem



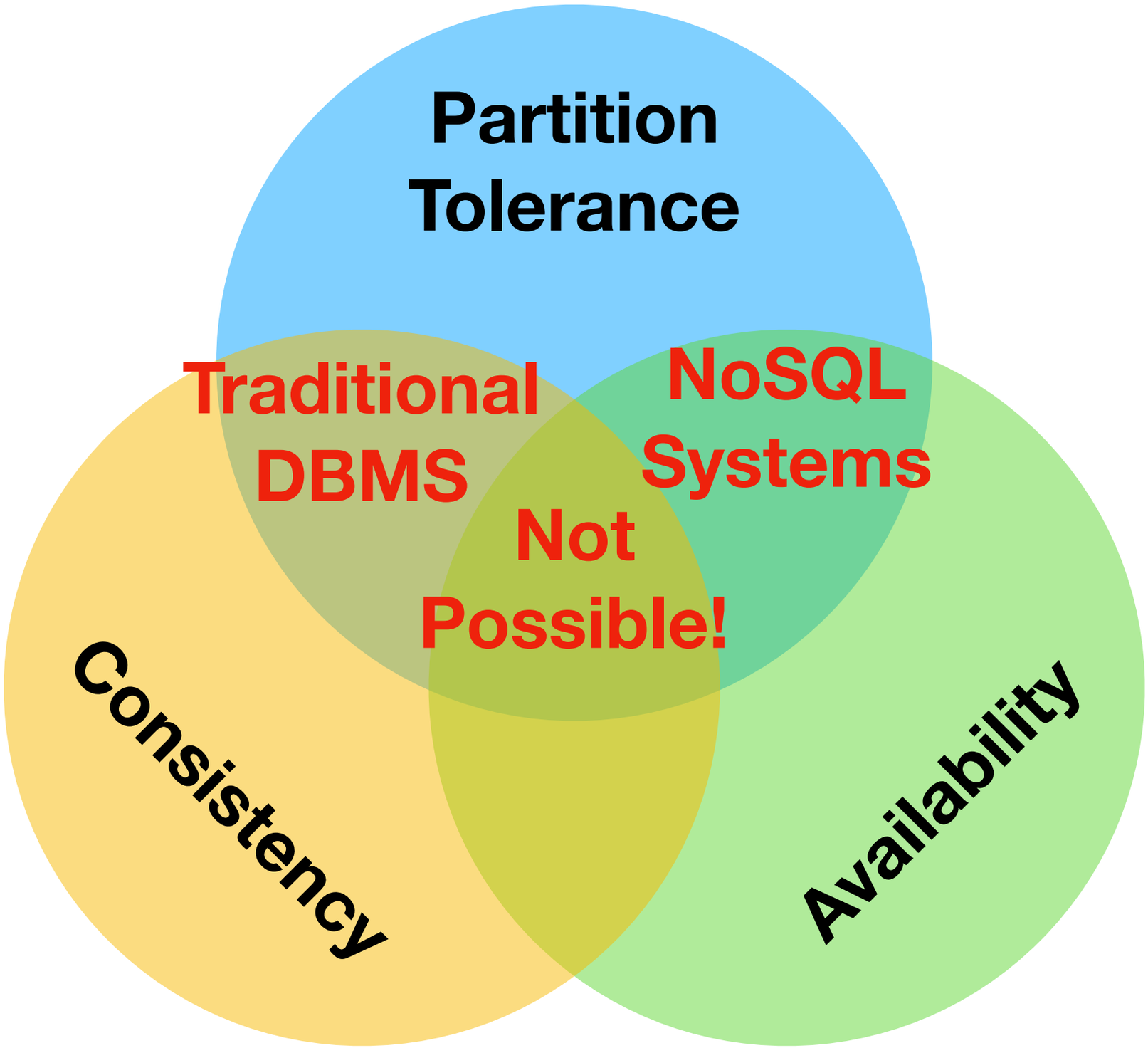
Slides by Immanuel Trummer, Cornell University

The CAP Theorem



Slides by Immanuel Trummer, Cornell University

The CAP Theorem



Slides by Immanuel Trummer, Cornell University

Eventual Consistency

- Traditional DBMS choose **consistency** over availability
- This is **not ideal** in scenarios such as online shopping
- Here, we want to be **available** at all costs
- Need to accept **inconsistency** according to CAP
- This inconsistency is **resolved eventually**

Slides by Immanuel Trummer, Cornell University

BASE Transactions

- **BASE** =
- **B**asically **A**vailable
- **S**oft State
- **E**ventually Consistent

NoSQL

- Systems that **move away** from traditional SQL DBMS
- **Broad** term covering many aspects such as
  - Reduced **consistency** (BASE)
  - Non-SQL query **languages**
  - Non-relational **data models**
  - ...

Slides by Immanuel Trummer, Cornell University

Apache Cassandra

- Distributed system, every node has the **same role**
- **Wide column** store ~ rows have different columns
- Supports **CQL**, simpler than SQL (e.g., no joins)
- Supports replication for **fault tolerance**
- Goal: **scale linearly** when adding new nodes
- Eventually consistent with **tunable consistency**

Slides by Immanuel Trummer, Cornell University

CAP Criticism

- Focuses on an **extreme case**: full partitions are rare
- **Simplifies tension** between conflicting design goals
- E.g., can decide A vs. C for single transactions
- E.g., consistency is not a binary property
- ...

NewSQL

- "**Traditional SQL**": ACID at the expense of performance
- **NoSQL**: give up ACID for higher performance
- **NewSQL**: new ideas for ACID with high performance

Slides by Immanuel Trummer, Cornell University

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates
- Modern **Hardware**
  - Main memory often fits entire DB
  - Distributed systems common

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates
- Modern **Hardware**
  - Main memory often fits entire DB
  - Distributed systems common

Design Implications ?

Slides by Immanuel Trummer, Cornell University

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates
- Modern **Hardware**
  - Main memory often fits entire DB
  - Distributed systems common

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates
- Modern **Hardware**
  - Main memory often fits entire DB → ***Main memory DBMS***
  - Distributed systems common

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates
- Modern **Hardware**
  - Main memory often fits entire DB → ***Main memory DBMS***
  - Distributed systems common → ***Replication for fault tolerance***

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates

→ ***Serial Execution! (Single Site Txs)***

- Modern **Hardware**
  - Main memory often fits entire DB → ***Main memory DBMS***
  - Distributed systems common → ***Replication for fault tolerance***

H-Store: Observations

- Modern Transaction **Workloads**
  - Short running transactions
  - No user input needed
  - Transactions ~ templates

→ ***Serial Execution! (Single Site Txs)***

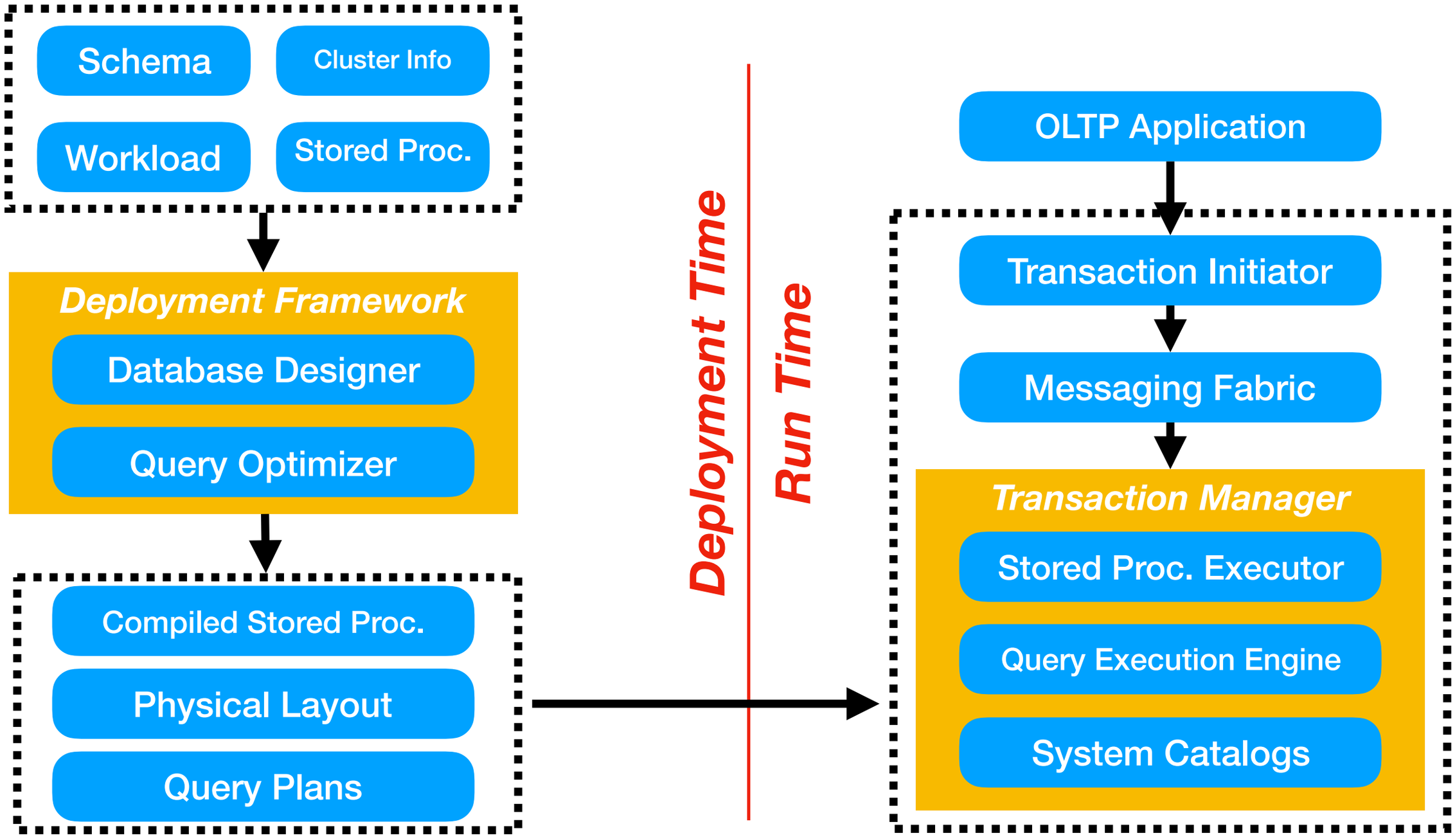
→ ***Exploit Pre-processing***

- Modern **Hardware**
  - Main memory often fits entire DB
  - Distributed systems common

→ ***Main memory DBMS***

→ ***Replication for fault tolerance***

H-Store Overview



H-Store: A High-Performance, Distributed Main Memory Transaction Processing System, R. Kallman et al.