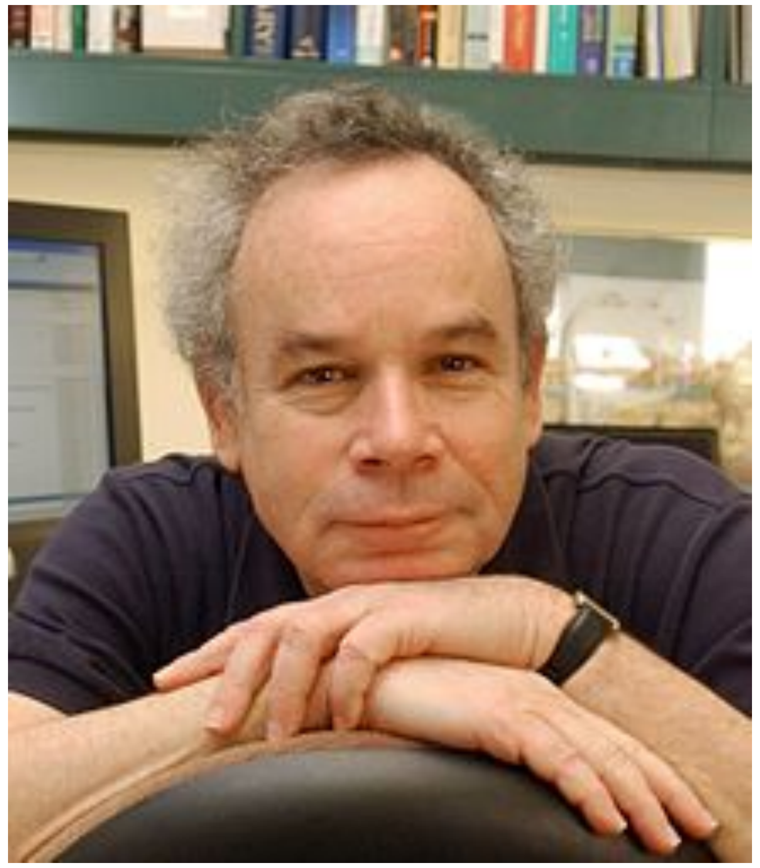
Database Systems: Achievements and Opportunities

Presented By: Lakshayy Dua (Data Engineer, HERE Maps)

ABOUT THE PAPER

- Publishing details:
    - Communications of the ACM (Association for Computing Machinery)
    - October 1991
    - Volume 34, No.10
- Hewlett-Packard Laboratories hosted the workshop supported by NSF (National Science Foundation) Grant
- Topic: “Background”
    - No descriptive algorithm or math included
    - Theoretical paper to convey 3 main ideas
- Message objective: “Database systems research is important and despite significant achievements it should be encouraged even further “

ABOUT THE AUTHORS



Avi Silberschatz

- Work has been cited over 23,000 times

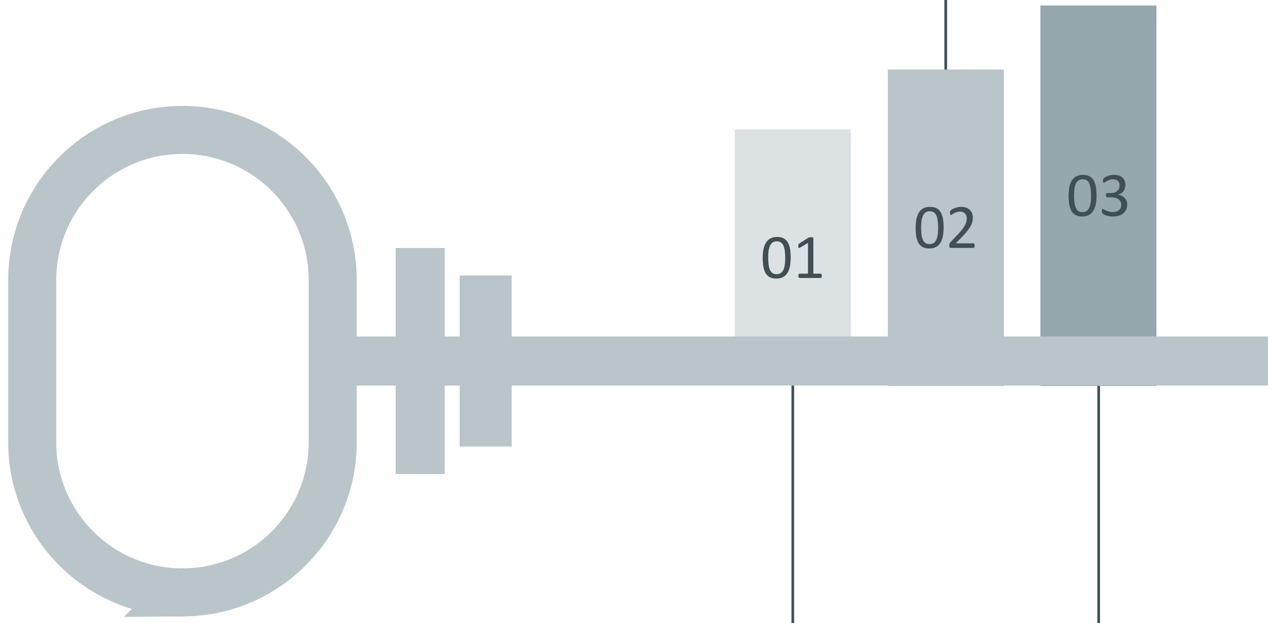
Michael Stonebraker

- Turing Award in 2014
- Known for contribution to Ingres, PostgreSQL and more

Jeff D. Ullman

- His textbooks on data structures, and databases are regarded as standards in their fields

3 IMPORTANT KEY IDEAS



Next Challenges

- Next Generation Applications
- New Kinds of Data
- Rule Processing
- Parallelism and more….

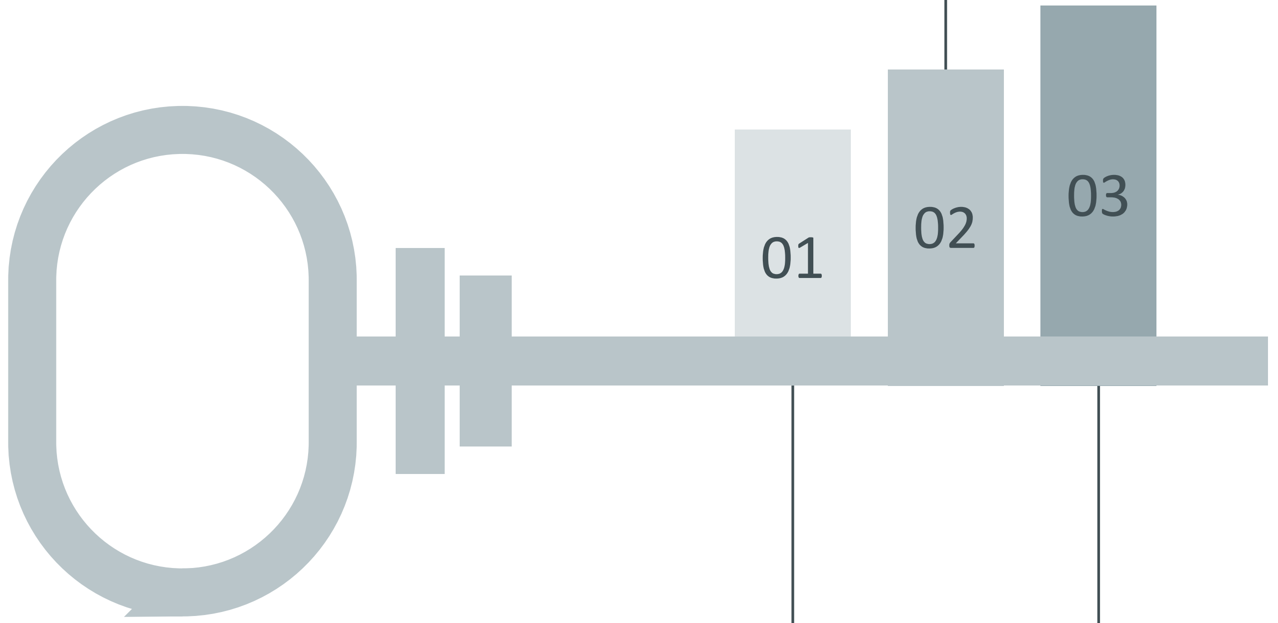
Achievements in 1970s & 1980s

- What's been done so far?
- What did we gain from it?
- Should we invest more money, time and effort?

Heterogenous Distributed DB Systems

- What's the need or application?
- What causes interoperability problem?
- How to address interoperability?

3 IMPORTANT KEY IDEAS



Achievements in 1970s & 1980s

- What's been done so far?
- What did we gain from it?
- Should we invest more money, time and effort?

Heterogenous Distributed DB Systems

- What's the need or application?
- What causes interoperability problem?
- How to address interoperability?

Next Challenges

- Next Generation Applications
- New Kinds of Data
- Rule Processing
- Parallelism and more….

ACHIEVEMENTS IN 20 YEARS (1970’s & 1980’s)

- Database research had gone in many different directions. But most prominent areas were:
    1. Relation Databases
    2. Transaction Management
    3. Distributed Databases

- Each offered radical simplification in dealing with data
- Improved capability to manage information

DATABASES IN EARLY 1970's

- In 1970's, 2 popular approach to construct a DBMS were:
  1. IBM's IMS (Information Management System) (1966)
  2. CODASYL (Conference on Data Systems Languages) (1969)
- Both tree-based and graph-based approaches have fundamental disadvantages:
  1. Must write a complex program to navigate through the database
  2. Application programs usually need to be rewritten, when database structure changes
- Lead to high costs for use and maintenance of databases

SOLUTION ?

RELATIONAL DATABASES (Pioneered by Edgar F. Codd)

- Simple tabular data structures (relations)
- Users access data through a high-level, nonprocedural (or declarative) query language
- Following concepts, contributed heavily to Relational Databases:
  1. High-Level Relational Query Languages
  2. Theory & Algorithms Necessary to Optimise Queries
  3. Algorithms to allocate tuples of relations to pages (blocks of records) in files on secondary storage
  4. Buffer Management Algorithms
  5. Indexing Techniques

RESULT: Because of 70's research we had Commercial RDMS Products in 80's

The first system sold as an RDBMS was Multics Relational Data Store (1978). Followed by Ingres and IBM BS12.

TRANSACTION MANAGEMENT

- Transaction is a sequence of operations that must appear "atomic" when executed
- To ensure a transaction transforms a database from one consistent state to another:
  1. Concurrency control to ensure each transaction appears to execute in isolation
  2. Transaction must execute in its entirety. ‘recovery’ provides assurance from h/w s/w failure

- Recovery is an important part of transaction management. Two major contribution to recovery:
  1. Write-ahead Logging
  2. Shadow-file Techniques
- Concurrency control algorithms were invented that ensure serializability:
  - 2 phase locking [growing phase, shrinking phase]
  - Timestamping accesses to prevent violations of serializability
  - Keeping multiple versions of data objects available

Logs can be used to roll forward the backup copy to the current state in case of failure

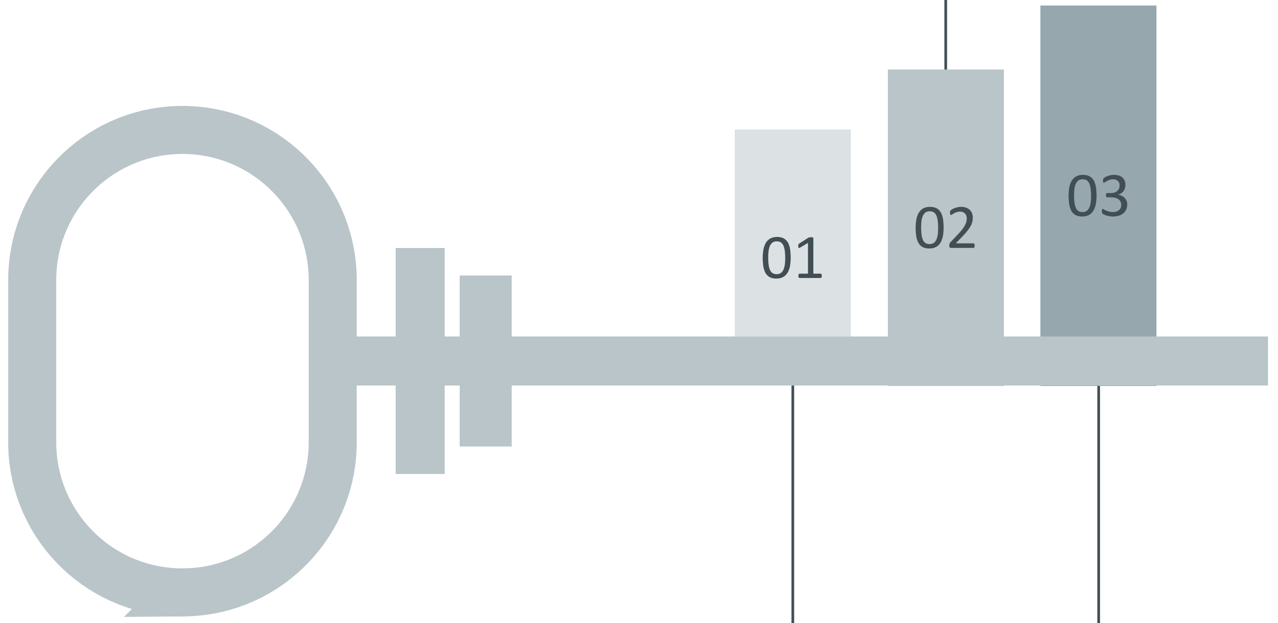
DISTRIBUTED DATABASES (Homogenous)

- Organizations are decentralized and require databases at multiple sites
- Distribution moves data closer to the people who are responsible for it and reduces remote communication costs
- If a single, central site goes down, all data is unavailable but not with distributed database
  - Example : (AWS RDS Multi A-Z), (GCP CloudSQL High Availability)
- In a multi-database environment we strive to achieve location transparency.
  - All data should appear as it’s located at one site

WHAT NEXT?

- By early 90’s the database research had paid off:
  - Both centralised and distributed relational DBMS have been commercialised
  - Object management ideas inspired by OOPs were expected to allow more general kinds of data elements to be placed in databases
  - Hardware was expected to become cheaper
- Q. Should we stop further research? A. NO!
  - Scale of future app is going to be different

3 IMPORTANT KEY IDEAS



Achievements in 1970s & 1980s

- What's been done so far?
- What did we gain from it?
- Should we invest more money, time and effort?

Heterogenous Distributed DB Systems

- What's the need or application?
- What causes interoperability problem?
- How to address interoperability?

Next Challenges

- Next Generation Applications
- New Kinds of Data
- Rule Processing
- Parallelism and more….

NEXT CHALLENGES

- Next Generation Application
- New Kinds of Data
- Rule Processing
- Scaling Up
- Parallelism

NEXT GENERATION APPLICATIONS

1. Databases are backbone of the CAD (computer aided design) systems.
   - Must maintain & integrate information about project from viewpoints of hundreds of subcontractors
   - Rules enforced in database can help calculate risk, ex: check risk before drilling
2. NASA is going to store 100 Petabytes of data in next few years
3. Insurance firms would start collecting data like images and video walkthroughs of properties

Convention DBMS cannot support such application.

NEW KINDS OF DATA

- Two Types of Data:
  1. Multimedia Data
  2. Scientific and Design Data (Very large arrays or sequences of data elements)

- What's required ?
  1. Ability to read a large object (LOB) only once and place it directly in final destination
  2. Protocols to chunk large objects into manageable size pieces for application to process
  3. Development of object-oriented databases

RULE PROCESSING

- Type of rules: declarative and imperative
- Imperative rule ex: Design change by one person should inform other person if his subsystem is affected

- Such rules may include:
  - elaborate constraints
  - triggered actions
  - complex deductions

- We can call such systems Knowledge-Base Systems

- What's required ?
  - Tools to validate and debug large collections of rules
  - Avoiding rule inconsistency to avoid repeated firing

NEXT CHALLENGES

- ✓ Next Generation Application
- ✓ New Kinds of Data
- ✓ Rule Processing
- o Scaling Up
- o Parallelism

SCALING UP

- 90’s DBMSs build a new index on a relation by locking it
- **New index** on big data may take days,
- **Data dump** of big database may take days,
  - Solution: Carry out both the operations incrementally (without making the data inaccessible)
- A new approach to backup and recovery in very large databases must be found for faster recovery of big data

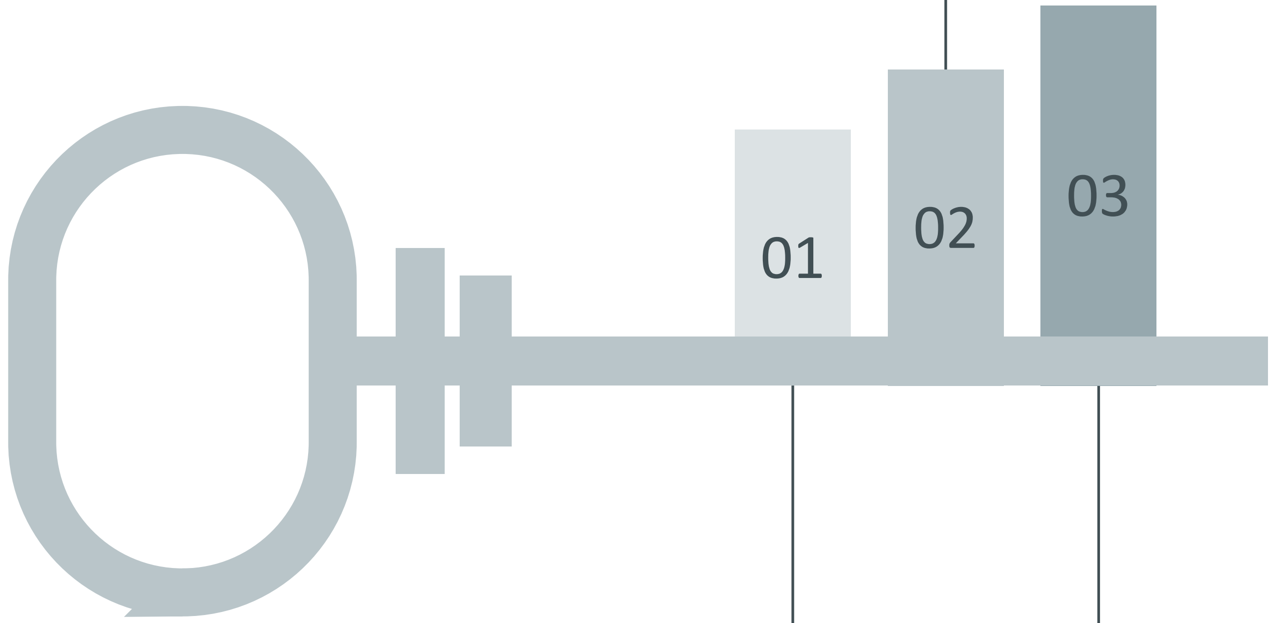
PARALLELISM

- Queries will take longer time for large databases
- Solution: Parallelizing a user command will allow it faster execution (but at a cost):
  - Take more computing resources – other users affected
  - System Thrashing (excessive paging operations)

NEXT CHALLENGES

- ✓ Next Generation Application
- ✓ New Kinds of Data
- ✓ Rule Processing
- ✓ Scaling Up
- ✓ Parallelism

3 IMPORTANT KEY IDEAS



Next Challenges

- Next Generation Applications
- New Kinds of Data
- Rule Processing
- Parallelism and more….

Achievements in 1970s & 1980s

- What's been done so far?
- What did we gain from it?
- Should we invest more money, time and effort?

Heterogenous Distributed DB Systems

- What's the need or application?
- What causes interoperability problem?
- How to address interoperability?

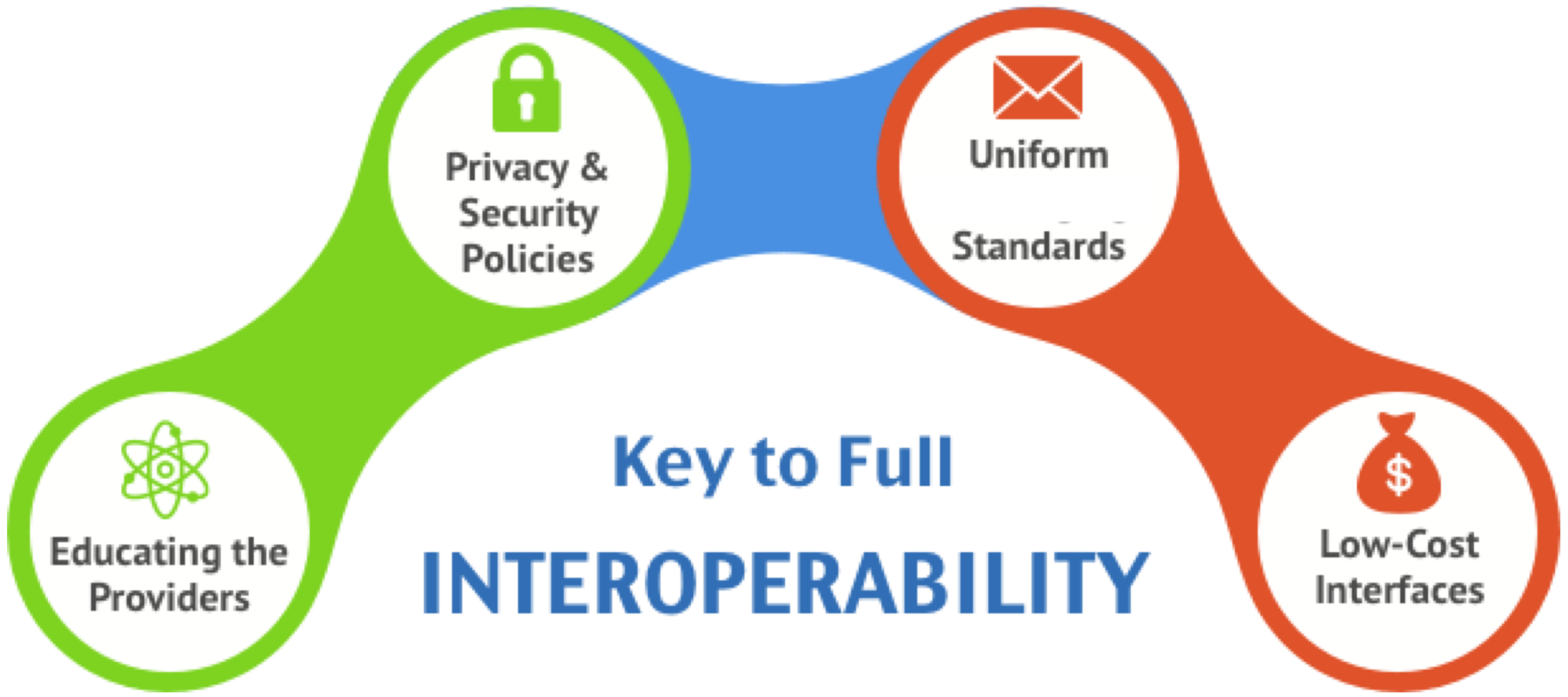
HETEROGENEOUS & DISTRIBUTED DATABASES

- There is one word wide web, one telephone network, what about one world database system?
- What's the need or application?
  1. Collaborative projects, example – human genome project
  2. General Example: Defense contractor has multiple subcontractors concerned with sub-projects

- Problem of making distributed heterogenous database behave as single database is called interoperability
- Problems faced in interoperability:
  1. Semantic Inconsistency
  2. Fusion of Evidence

HOW TO ADDRESS INTEROPERABILITY?

- Browsing
- Mediators
- Name Services
- Security
- Transaction Management



BROWSING

- **Assumption:**

  We have a universal query language that can access individual or merged view of collection of database.

- **Situation:**

  If an inconsistency is detected, or if missing information appears to invalidate a query.

- There must be some system for explaining to the user how the data arrived in that state and, in particular, from what databases it was derived.

- **Solution:** BROWSING
  1. Ability to interrogate the structure of the database
  2. And when multiple databases are combined, allows interrogating nature of process that merges data

MEDIATORS

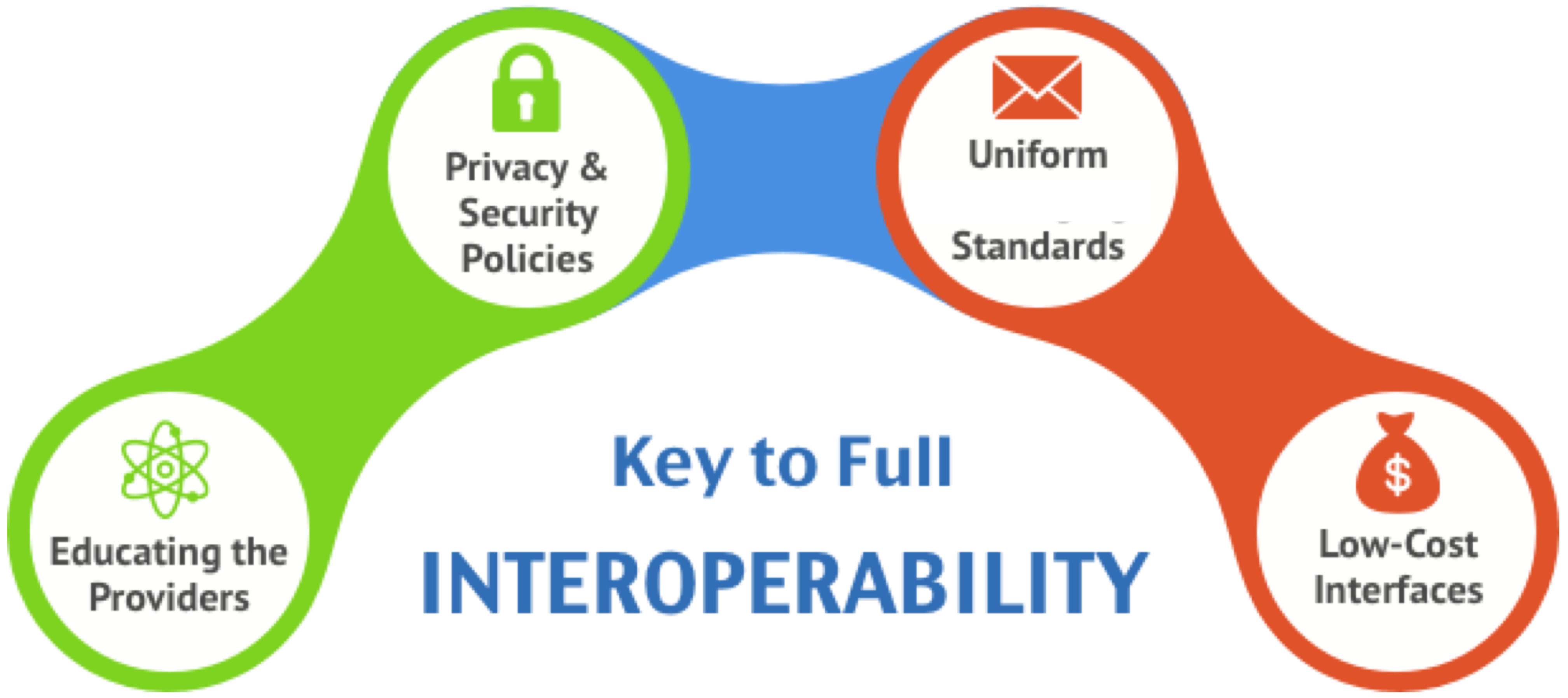
- Semantic inconsistency and fusion problems are severe
- We need a class of information resources that stand between user & heterogeneous databases
- Example: CS PhD mediator, Library mediator
- Some agreement regarding language and model standards

NAME SERVICES

- NSF program manager must be able to consult a national name service to discover the location and name of the databases or mediators of interest
- User must be able to discover the existence of relevant data sets
- Mechanism by which items enter and leave such name servers and organization of such systems is an open issue

HOW TO ADDRESS INTEROPERABILITY?

- ✓ Browsing
- ✓ Mediators
- ✓ Name Services
- o Security
- o Transaction Management



SECURITY

- A heterogeneous, distributed database system will need to cope with a world of multiple authenticators of variable trustworthiness
- Problems:
    1. A widely distributed system may have millions of users
    2. A given user may be identified differently on different systems
    3. Access permission might sometimes be based on role

- Solution: Having intervening sites
    1. Acts as intermediate agents for users
    2. Data may pass through and be manipulated by these intervening sites

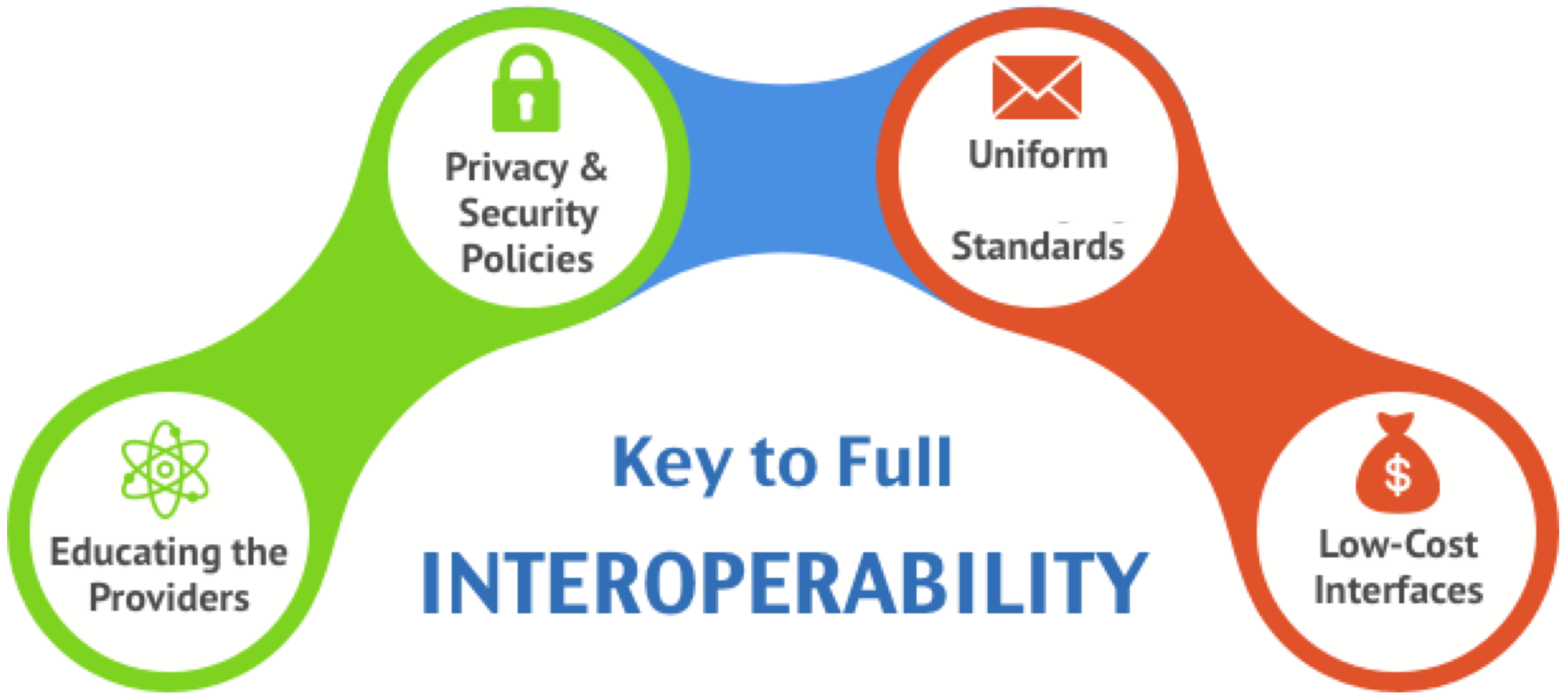
TRANSACTION MANAGEMENT

- Difficult because:
    - Each local DBMS may be using a different type of a concurrency control scheme
    - Situation becomes worse if we wish to preserve local autonomy of each local databases and allow local and global transactions to execute in parallel

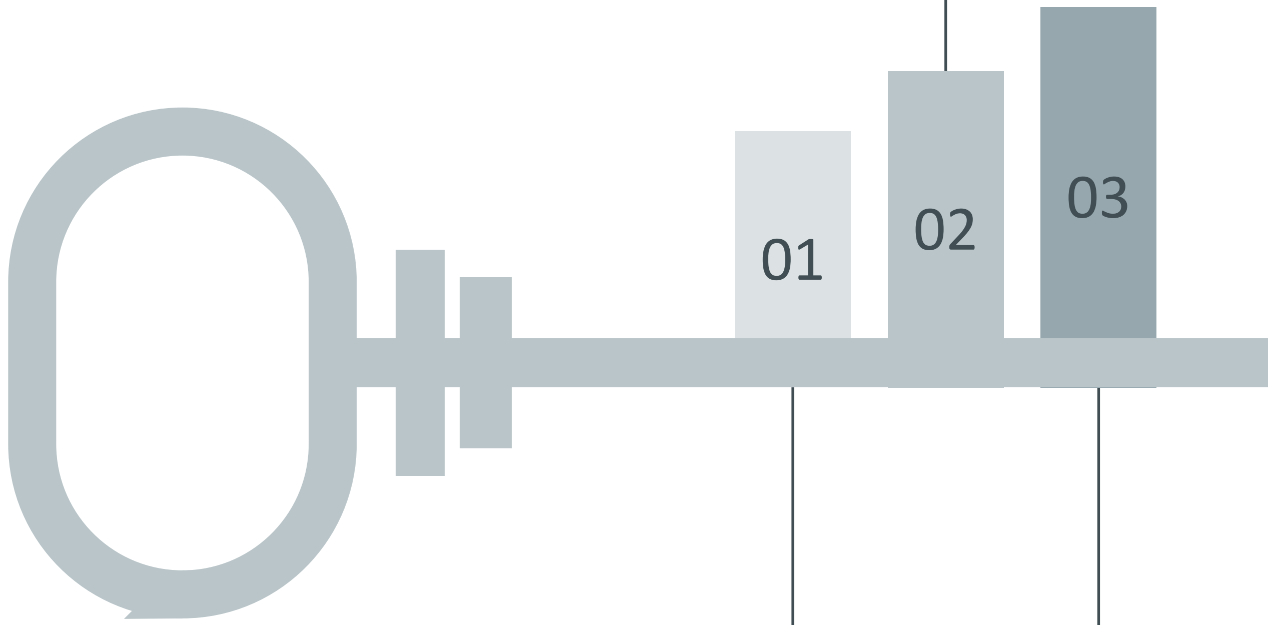
- Solution:
    - Restrict global transactions to retrieve-only access
    - Explore reliable transaction management, where global and local transactions are allowed to both read and write data

HOW TO ADDRESS INTEROPERABILITY?

- ✓ Browsing
- ✓ Mediators
- ✓ Name Services
- ✓ Security
- ✓ Transaction Management



3 IMPORTANT KEY IDEAS



Next Challenges

- Next Generation Applications
- New Kinds of Data
- Rule Processing
- Parallelism and more….

Achievements in 1970s & 1980s

- What's been done so far?
- What did we gain from it?
- Should we invest more money, time and effort?

Heterogenous Distributed DB Systems

- What's the need or application?
- What causes interoperability problem?
- How to address interoperability?

KEY TAKEAWAYS

1. Next generation application will be very different and require rethinking or extension of current concepts

2. Cooperation between different organizations on common problems will require heterogeneous, distributed databases.

3. Future database systems research offers:
   - New intellectual challenges
   - Products and technology of great social and economic welfare. ex – medicine field

THANK YOU!

ANY QUESTIONS?