SQDATA
CORPORATION

IMS to IDAA

Keeping Your Data Current

Prepared for the:

IMS Tech Symposium

17 March 2015

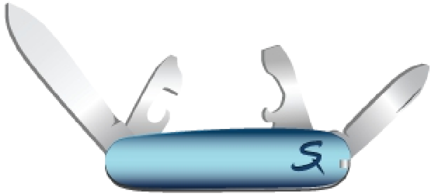
Agenda

- Introduction
- IDAA Overview
- Replicating IMS to IDAA
- Design Considerations
- Q & A
- Conclusion

About the Speaker

- **Scott Quillicy**
  - ✔ 30+ Years Database Experience
  - ✔ Commercial Database Software Development
  - ✔ Deployment of Complex Data Integration Solutions
- **Founded SQData to Provide Customers with:**
  - ✔ An Enterprise Class Data Integration / Replication Framework
  - ✔ A Solution that Handles Both Relational and Non-Relational Data
  - ✔ Technology Built Around Best Practices
- **Specialization**
  - ✔ Database Replication
  - ✔ IMS → the More Complex, the Better
  - ✔ Heterogeneous Database Integration
  - ✔ Continuous Availability
  - ✔ Database Performance

About SQData



- **“Swiss Army Knife of Data Integration Tools”**
- **Core Competencies**
  - High-Performance Changed Data Capture (CDC)
  - Non-Relational Data → IMS, VSAM, Flat Files
  - Relational Databases → DB2, Oracle, SQL Server, etc.
  - Deployment of Complex Data Integration Solutions
  - Continuous Availability of Critical Applications
  - Data Conversions / Migrations
- **Customer Usage**
  - Relational and Non-Relational Data
  - Data Replication – Relational and Non-Relational
  - ETL (Bulk Data Extracts/Loads)
  - Application Integration
  - Business Event Publishing
  - Data Conversions / Migrations

Why Replicate IMS to IDAA?

- Provide a Method of Analyzing Data Outside of IMS
- Real-Time Business Intelligence / Advanced Analytics
- Combine with Data from other Applications
- Save Significant CPU Cycles for Intense Queries
- Compliment Established Data Warehouse(s)

Best Practices Summary

- **Let the Business Drive the Effort**
  - Ensures Proper Alignment with Business Goals
  - Queries Drive the Data Model Design
  - Avoid I/T Initiated 'Build it and They will Come'
- **Temper the Exuberance**
  - Inevitable After Successful Implementation for a Given Application
- **Align with Enterprise Data**
  - Where I/T Comes Takes a Lead Role
  - Existing Data Warehouse / Business Intelligence Setups
  - Infrastructure / Data Integration
- **Use an Iterative Approach for Implementation**
  - Agile / Agile Like
  - Set the Relational Mindset Aside
  - Allows for 'Adjustments' without Major Schedule Impact

What is IDAA?

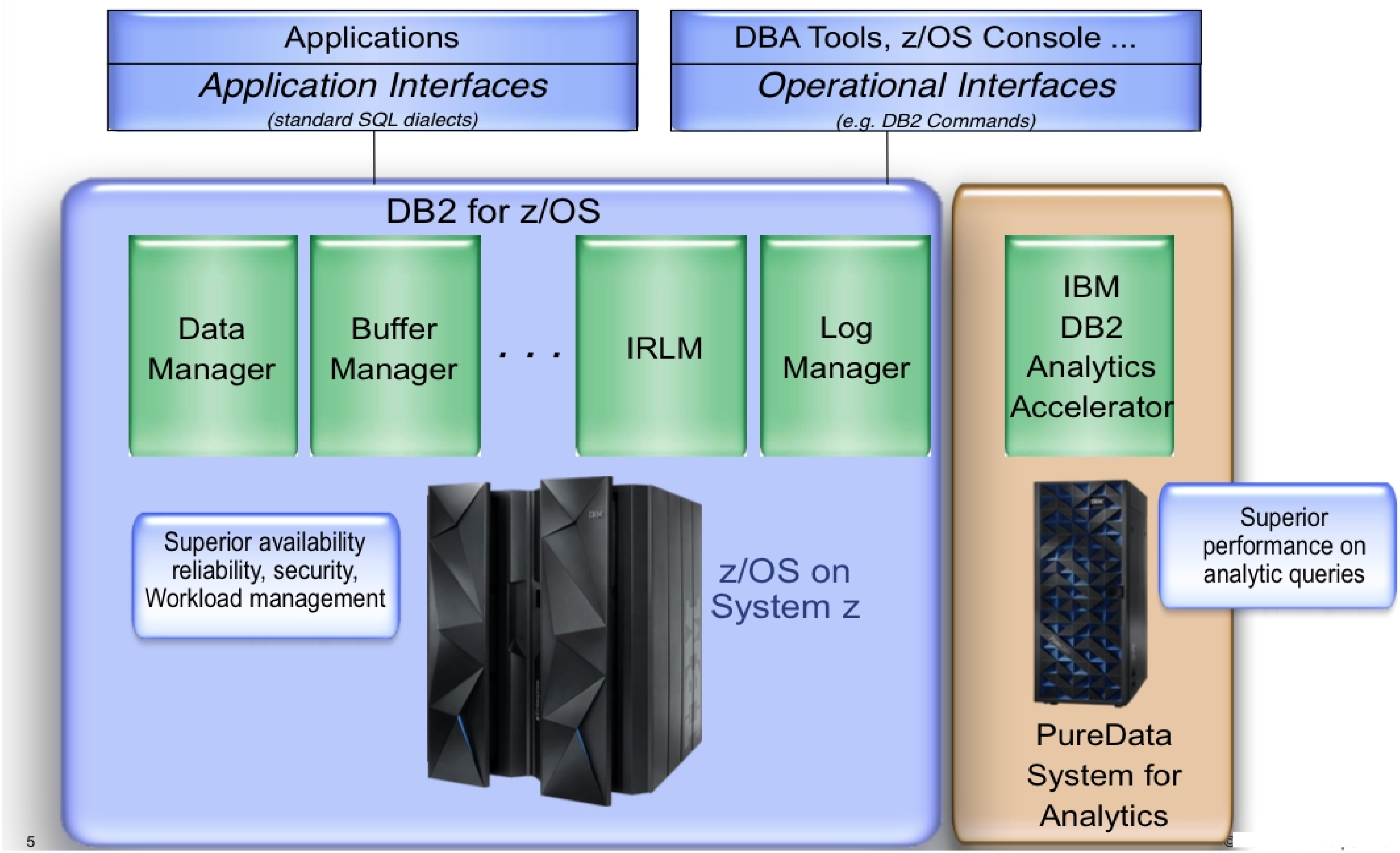
- Netezza Appliance Coupled to DB2
- Minimal Application Changes Required

The hybrid computing platform on zEnterprise

- *Supports transaction processing and analytics workloads concurrently, efficiently and cost-effectively*
- *Delivers industry leading performance for mixed workloads*

DB2 Analytics Accelerator and DB2 for z/OS

IDAA Architecture



Applications

Application Interfaces

(standard SQL dialects)

DBA Tools, z/OS Console ...

Operational Interfaces

(e.g. DB2 Commands)

DB2 for z/OS

Data Manager

Buffer Manager

. . .

IRLM

Log Manager

IBM DB2 Analytics Accelerator

Superior availability reliability, security, Workload management

z/OS on System z

Superior performance on analytic queries

PureData System for Analytics

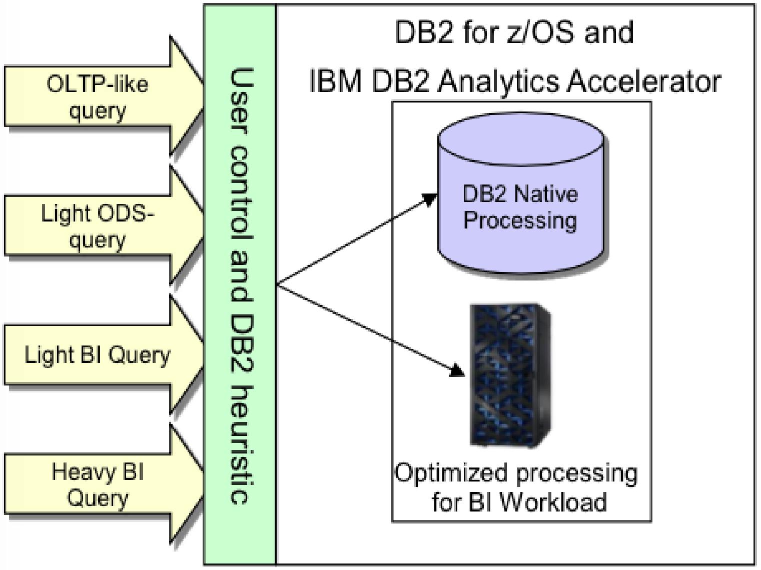
5

Running Queries on IDAA

- **Special Register** → CURRENT QUERY ACCELERATION
- **Mandatory Criteria**
  - ✔ All Tables in Query (Join) Exist in the IDAA
  - ✔ Query is Defined as Read Only
  - ✔ Cursor is Not Defined as Scrollable
  - ✔ Row Set Cursor is Not Remote
  - ✔ Query is from a Package and Not a Plan with DBRMs
  - ✔ Query is a SELECT or INSERT FROM SELECT Statement
- **SQL Restrictions**
  - ✔ Special Registers Other Than:
    - CURRENT DATE
    - CURRENT TIME
    - CURRENT TIMESTAMP
  - ✔ Sequence Expressions → NEXTVAL or PREVVAL
  - ✔ User Defined Functions (UDFs)
  - ✔ MIN / MAX with Strings or More than Four (4) Arguments

Query Routing

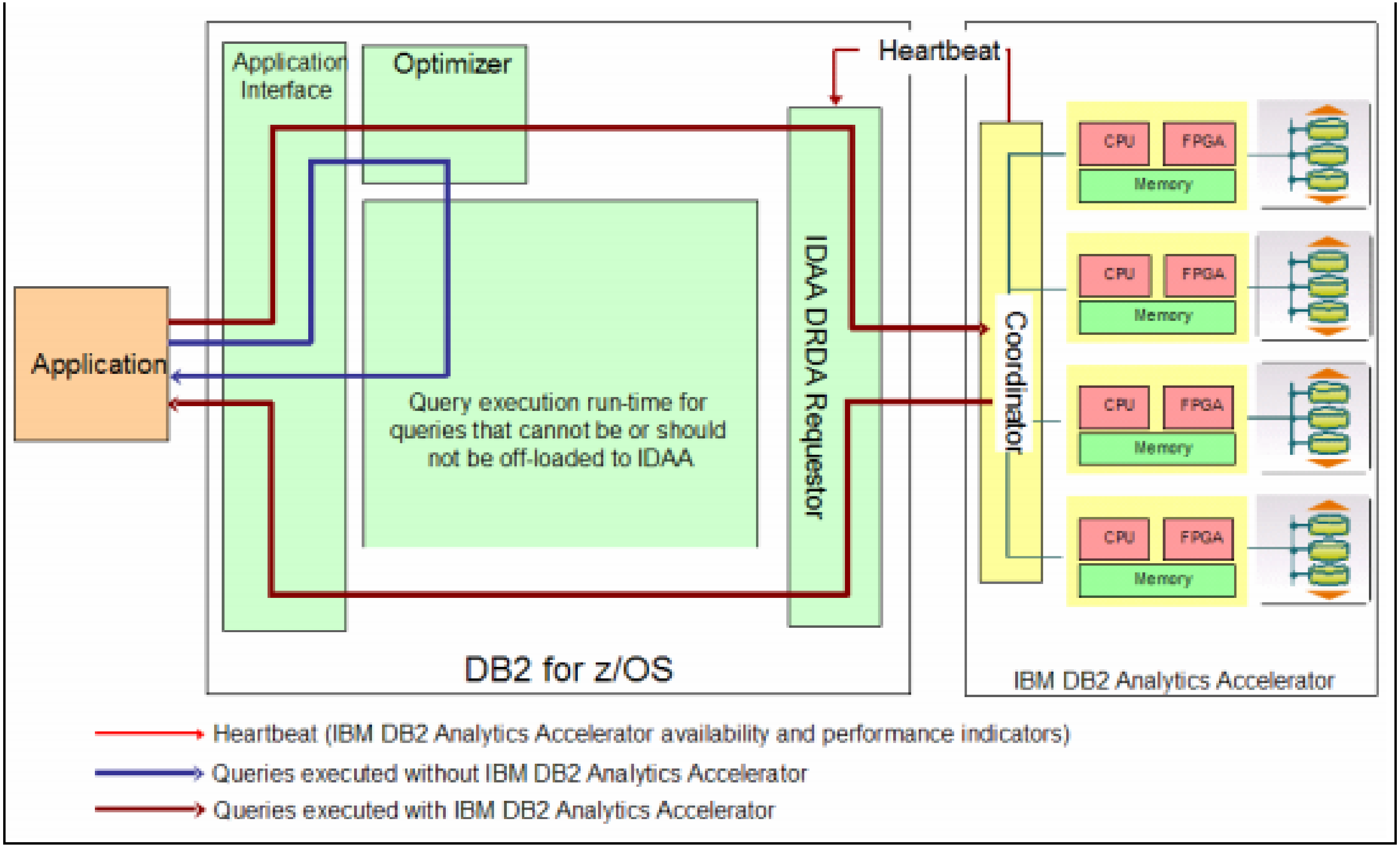
- Values for CURRENT QUERY ACCELERATION


DB2 for z/OS and
IBM DB2 Analytics Accelerator
OLTP-like query
Light ODS-query
Light BI Query
Heavy BI Query
User control and DB2 heuristic
DB2 Native Processing
Optimized processing for BI Workload

- Single and unique system for mixed query workloads
- Dynamic decision for most efficient execution platform
- New special register QUERY ACCELERATION
- New heuristic in DB2 optimizer

Value	Description
NONE	No query is routed to the accelerator
ENABLE	A query is routed to the accelerator if it satisfies the acceleration criteria including the cost and heuristics criteria. Otherwise it is executed in DB2. If there is an accelerator failure while running the query, or the accelerator returns an error, DB2 will return a negative SQL Code to the application
ENABLE WITH FAILBACK	A query is routed to the accelerator if it satisfies the acceleration criteria including the cost and heuristics criteria. Otherwise it is executed in DB2. Under certain conditions the query will run on DB2 after it fails in the accelerator. In particular any negative SQL code will cause failback to DB2 during PREPARE or first OPEN. No failback is possible after a successful OPEN of a query
ALL	A query is routed to the accelerator, if it cannot execute the query fails and a negative return code is passed back to the application
ELIGIBLE	A query is routed to the accelerator if it satisfies the acceleration criteria irrespective of the cost and heuristics criteria. Otherwise it is executed in DB2

Query Execution Flow


Application
Application Interface
Optimizer
Heartbeat
IDAA DRDA Requestor
Query execution run-time for queries that cannot be or should not be off-loaded to IDAA
DB2 for z/OS
Coordinator
CPU
FPGA
Memory
IBM DB2 Analytics Accelerator

Heartbeat (IBM DB2 Analytics Accelerator availability and performance indicators)

Queries executed without IBM DB2 Analytics Accelerator

Queries executed with IBM DB2 Analytics Accelerator

Key Considerations

- **Golden Rule** → Good Distribution = Good Performance
- **Objective**
  - ✔ All Tables are Distributed Across All Active Database Blades
  - ✔ All Queries Run Parallel Against All Active Database Blades
  - ✔ All Loads Run Parallel Against All Active Database Blades
- **Distribution Key Selection**
  - ✔ Primary Key of Source Data
  - ✔ Columns Used for Joins
  - ✔ Columns with High Cardinality
  - ✔ Columns Frequently Aggregated On
- **Organizing Key Selection**
  - ✔ Only Use on Tables with > 1M Rows
  - ✔ One or More Columns of Primary Key → Incremental Update Performance
  - ✔ Columns Used as Common Predicates

Inserts, Updates and Deletes in IDAA

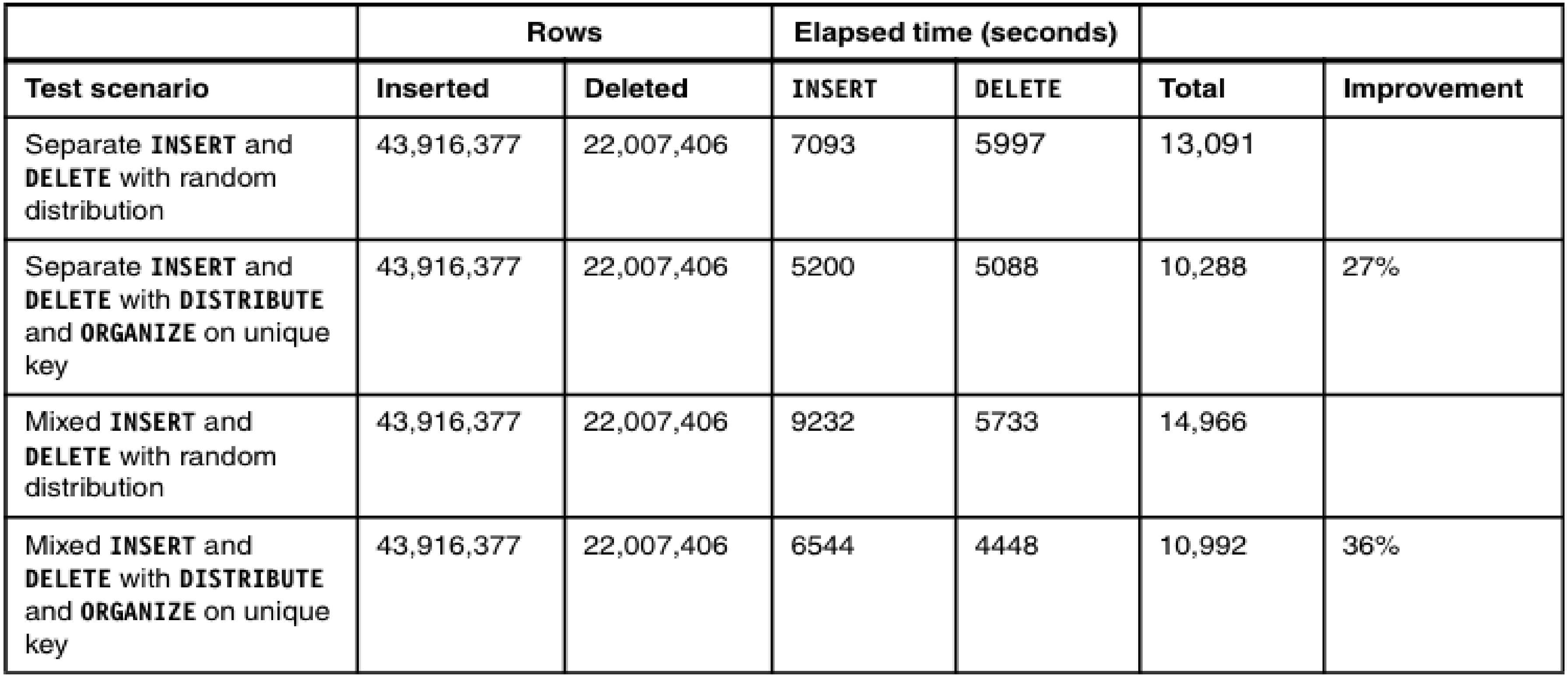
- **Updates → Delete / Insert Pairs**

- **Inserts**
  - ✔ Appends Data to End of File → Very Fast
  - ✔ Speed is Based on Number of Rows Being Inserted

- **Deletes**
  - ✔ Must Scan Entire File
  - ✔ Select Row – Update with Delete Flag
  - ✔ Replace Row in Place
  - ✔ Groom Process Cleans up Files
  - ✔ Speed is Based on Size of File and Number of Rows Being Deleted

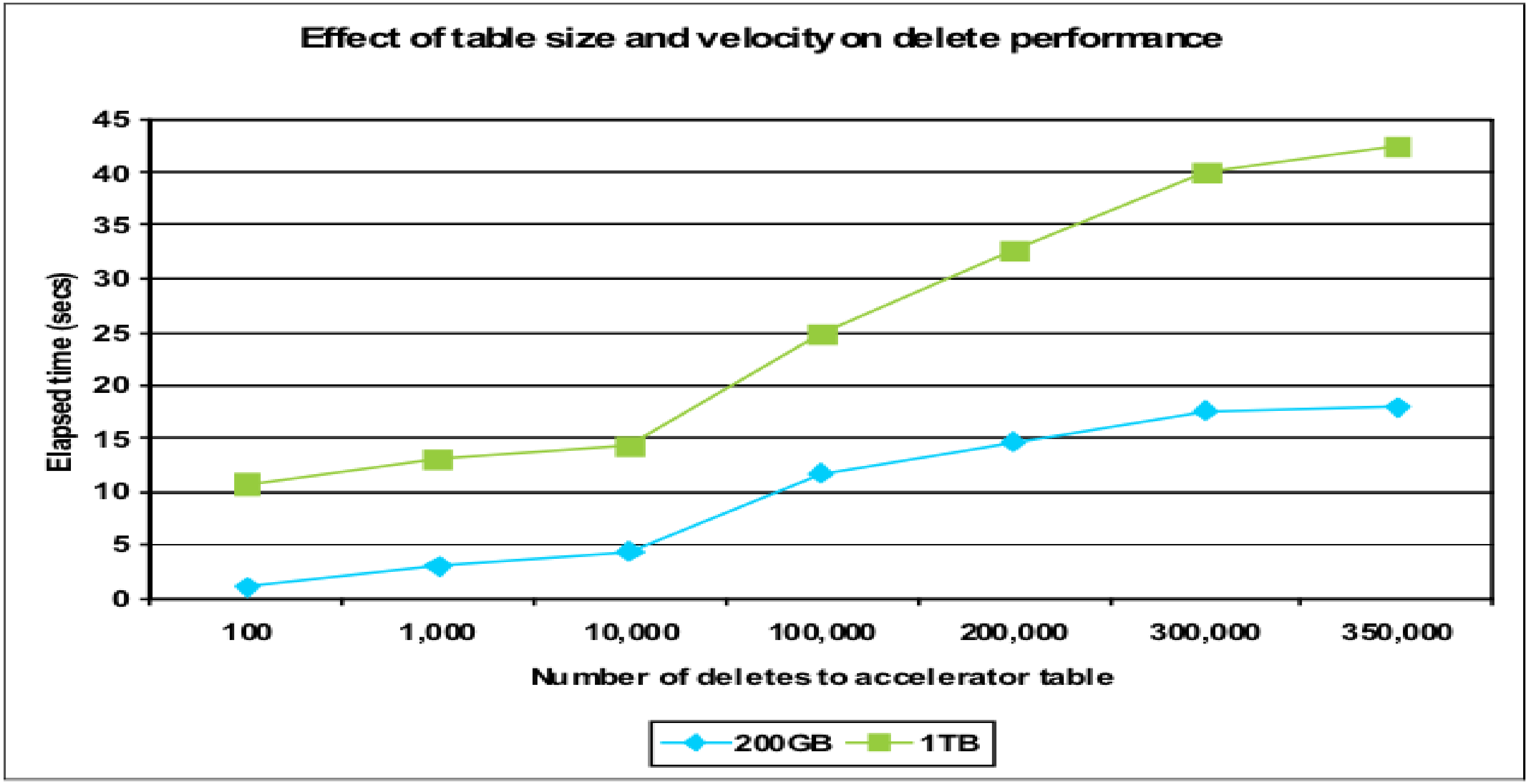
Insert / Delete Performance

- With and Without Distribution and Organizing Keys



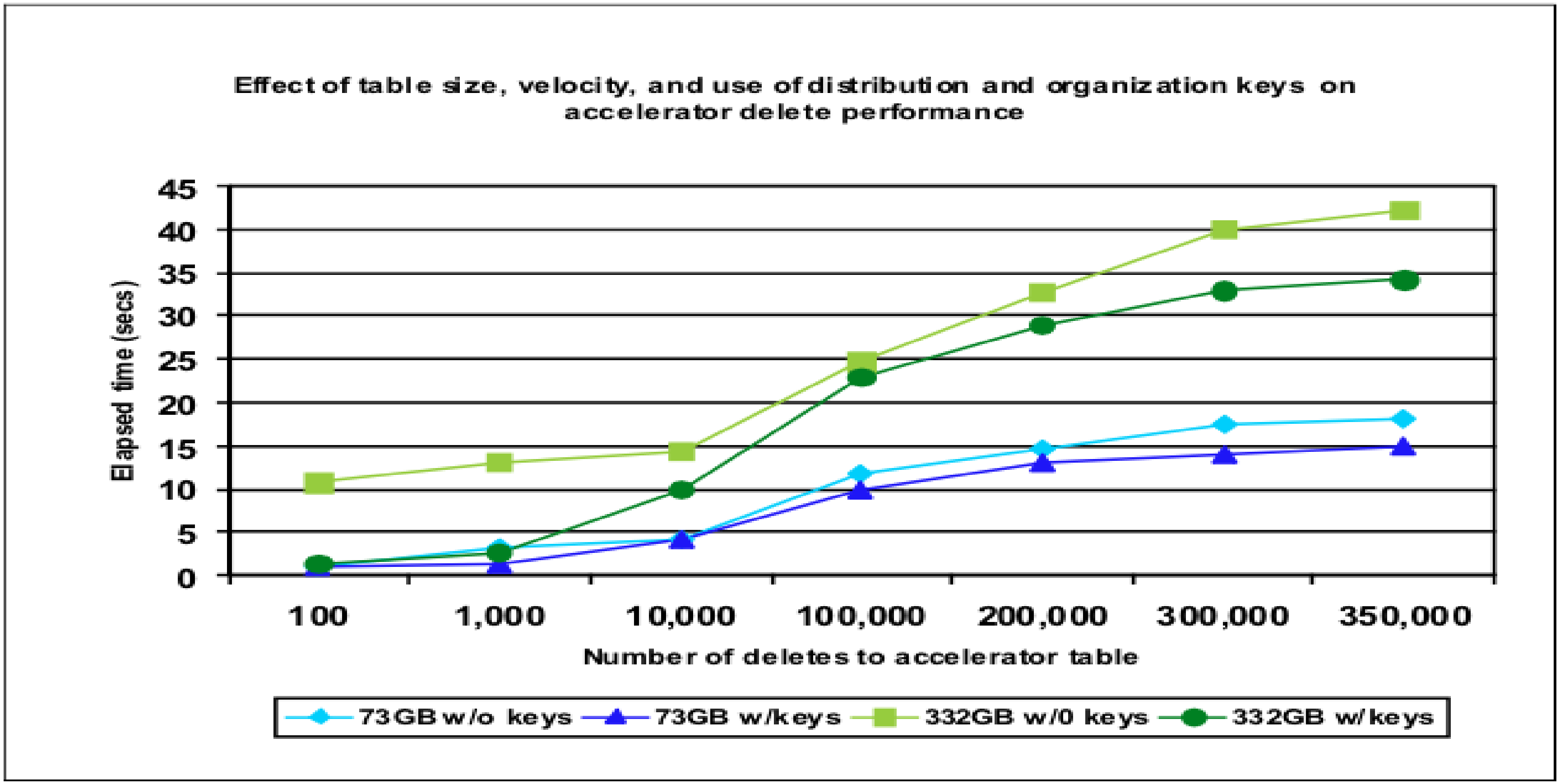
	Rows		Elapsed time (seconds)			
Test scenario	**Inserted**	**Deleted**	**INSERT**	**DELETE**	**Total**	**Improvement**
Separate **INSERT** and **DELETE** with random distribution	43,916,377	22,007,406	7093	5997	13,091	
Separate **INSERT** and **DELETE** with **DISTRIBUTE** and **ORGANIZE** on unique key	43,916,377	22,007,406	5200	5088	10,288	27%
Mixed **INSERT** and **DELETE** with random distribution	43,916,377	22,007,406	9232	5733	14,966	
Mixed **INSERT** and **DELETE** with **DISTRIBUTE** and **ORGANIZE** on unique key	43,916,377	22,007,406	6544	4448	10,992	36%

Delete Performance → Table Size



© Copyright IBM Corporation

Delete Performance → Table Size and Keys



© Copyright IBM Corporation

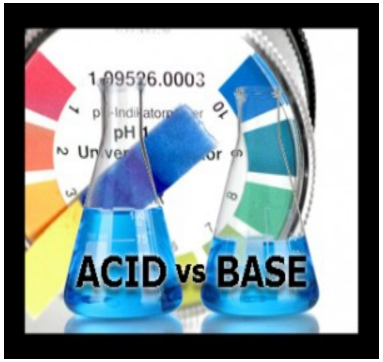
ACID vs BASE

- **ACID** → Properties Guarantee DB Transactions are Processed Reliably
  - ✔ **A**tomicity → All or Nothing...either the Transaction Commits or it Doesn't
  - ✔ **C**onsistency → Transaction brings DB from One Valid State to Another
  - ✔ **I**solation → Concurrency
  - ✔ **D**urability → Once a Transaction Commits, it Remains Committed

- **BASE** → Eventual Consistency
  - ✔ **B**asically **A**vailable → Data is There...No Guarantees on Consistency
  - ✔ **S**oft State → Data Changing Over Time...May Not Reflect Commit Scope
  - ✔ **E**ventual Consistency → Data will *Eventually* become Consistent

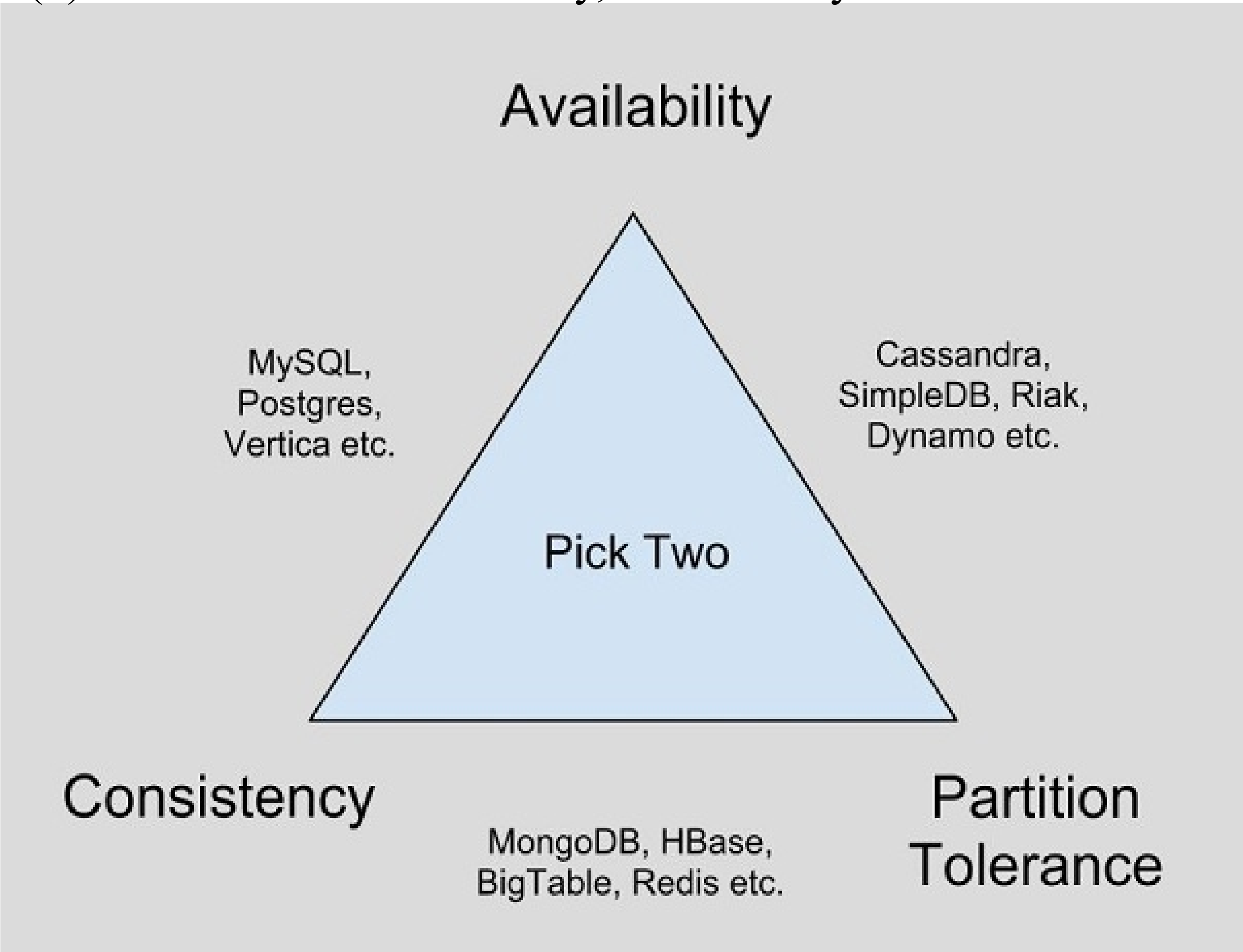
More Info: Charles Rowe – Shifting pH of Database Transaction Processing

http://www.dataversity.net/acid-vs-base-the-shifting-ph-of-database-transaction-processing/



CAP Theorem

- Eric Brewer - 1998 → Impossible for a Distributed System to Provide All Three (3) Guarantees of Availability, Consistency and Partition Tolerance


Availability
MySQL, Postgres, Vertica etc.
Cassandra, SimpleDB, Riak, Dynamo etc.
Pick Two
Consistency
MongoDB, HBase, BigTable, Redis etc.
Partition Tolerance

Agenda

- Introduction
- IDAA Overview
- **Replicating IMS to IDAA**
- Design Considerations
- Q & A
- Conclusion

Getting Started

- **Verify Design Meets Business Requirements**
- **Create Target Table(s) in DB2**
  - ✓ May be Empty
  - ✓ Needs to be Registered in the DB2 Catalog
  - ✓ Allocate a Small Tablespace if Data Will Not Reside in DB2
- **Create Target Table(s) in IDAA**
  - ✔ Add Table Wizard in Accelerator Studio
  - ✔ Stored Procedure → SYSPROC.ACCEL_ADD_TABLES
- **Setup Initial Load Interfaces**
  - ✔ Typically Run Once
  - ✔ Can be Used in Place of CDC for Certain Tables
- **Setup CDC Interfaces**
  - ✓ Determine Latency Settings
  - ✓ Group for Consistent and/or Eventually Consistent Replication

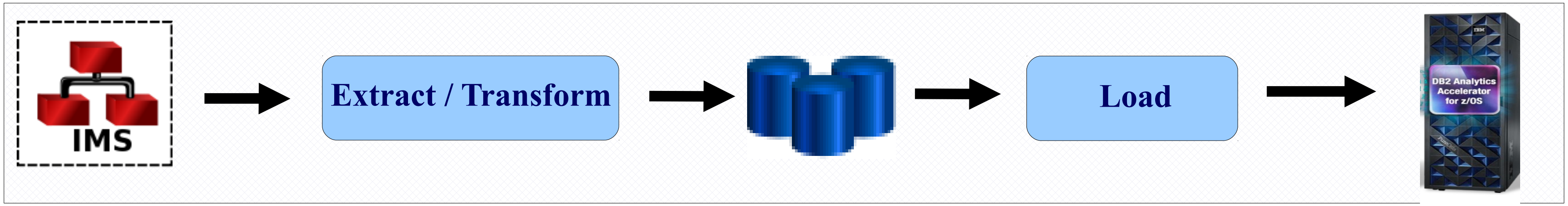
High-Risk Elements

- **No Access to Subject Matter Expert(s)**
  - ✔ Significantly Increases Risk
  - ✔ Extends the Project Timeline
  - ✔ Results in Guesswork for Design and Mapping
- **Underestimating the Complexity of IMS to Relational**
- **Big Bang Approach - Attempting to Attack Everything at Once**
  - ✔ Recommend Phased Implementations
  - ✔ Subsequent Migrations become Shorter: Experience & Lessons Learned
- **Fast-Tracking Planning and Analysis**
  - ✔ Causes Unnecessary Rework and Waste
  - ✔ More Time Spent on the Front End Saves on the Back End
- **High-Transaction Workload on the IMS Side**
  - ✔ Applies Primarily to Application Conversion
  - ✔ Performance will NOT be the Same as with IMS

The Role of ETL and CDC

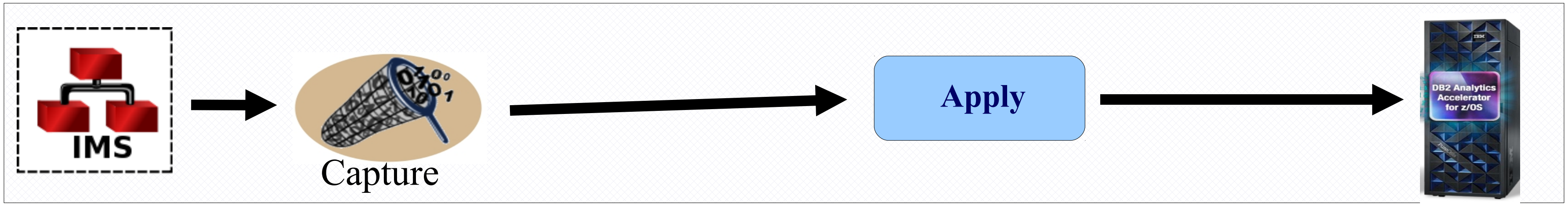
ETL (Extract, Transform, Load):

- ✓ Full Data Extract / Load
- ✓ Data Transformation Logic Defined in this Step
- ✓ Iterative Process – Must be Fast and Efficient
- ✓ Should Minimize Data Landing



CDC (Changed Data Capture):

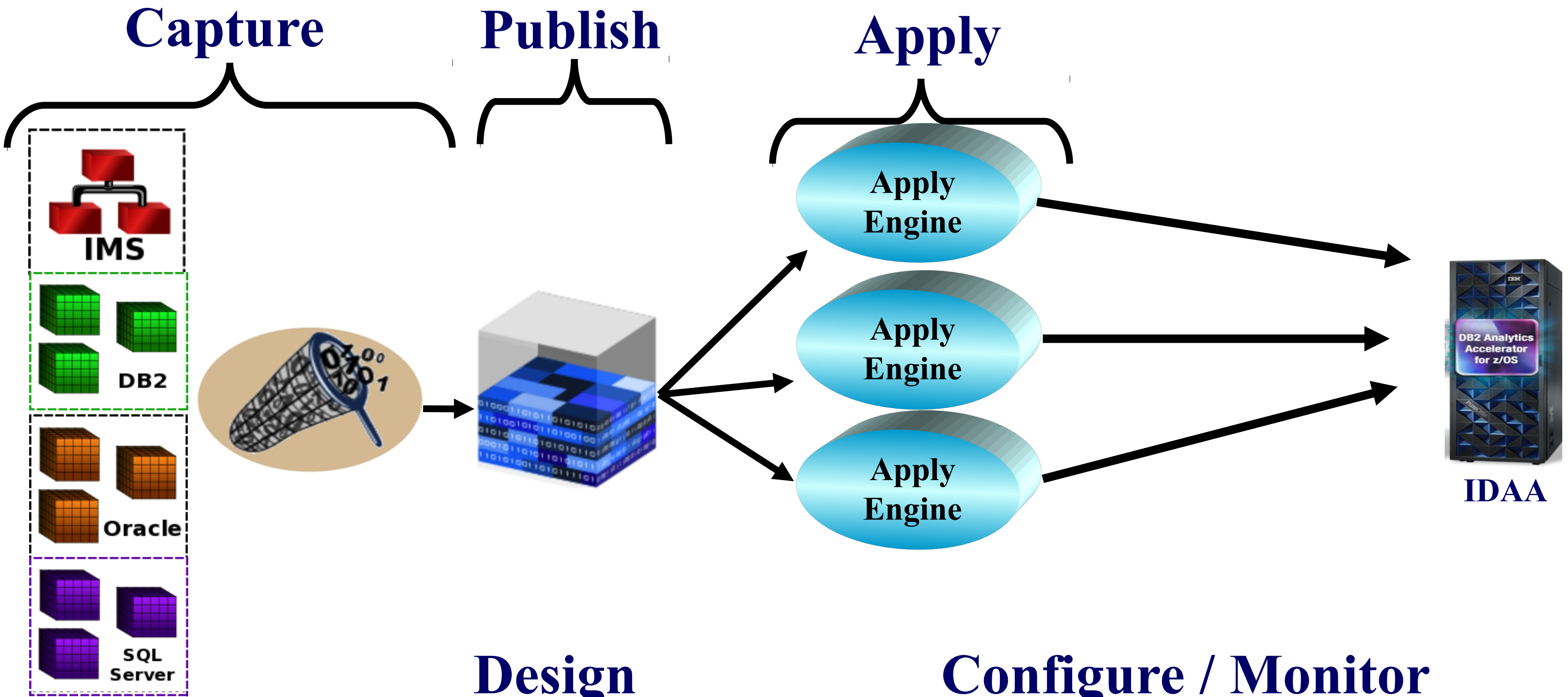
- ✓ Move Only Data that has Changed
- ✓ Re-Use Data Transformation Logic from ETL
- ✓ Near-Real-Time / Deferred Latency

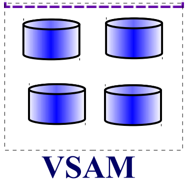


Performing the Initial Load

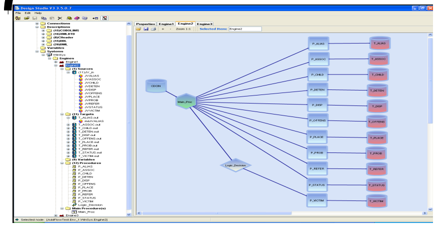
- Transformation / Mapping Logic Done Here → Reused in CDC
- Should be Able to Run Against Live Sources
- Make Sure to Truncate Before Loading → Otherwise Duplicates
- May be Used in Leu of Incremental Updates (CDC)
- **NZLOAD**
  - ✓ Native Netezza Loader
  - ✓ Operates the Same in IDAA and Netezza Only Environments
- **IDAA Loader**
  - ✓ IBM Product Offering
  - ✓ Allows Simultaneous Loading into DB2 and IDAA
  - ✓ Allows IDAA Only Loads

SQData IDAA Replication Ecosystem

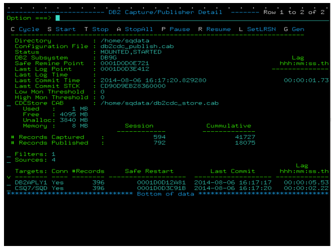




Design



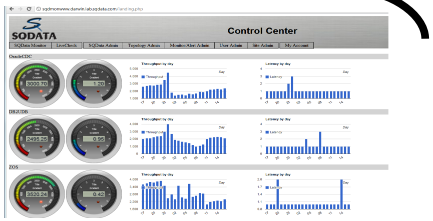
Configure / Monitor



```
------------------------- DB2 Capture/Publisher Detail  -------- Row 1 to 2 of 2
Option ===>

 C Cycle  S Start  T Stop  A StopAll  P Pause  R Resume  L SetLRSN  G Gen
 ----------------------------------------------------------------------------
  Directory          : /home/sqdata
  Configuration File : db2cdc_publish.cab
  Status             : MOUNTED,STARTED
  DB2 Subsystem      : DB9G                                              Lag
  Safe Remine Point  : 0001D0D0E721                               hhh:mm:ss.th
  Last Log Point     : 0001D0D3E412                              -------------
  Last Log Time      :
  Last Commit Time   : 2014-08-06 16:17:20.829280                  00:00:01.73
  Last Commit STCK   : CD90D9EB28360000
  Low Mon Threshold  : 0
  High Mon Threshold : 0
_ CDCStore CAB       : /home/sqdata/db2cdc_store.cab
    Used    :    1 MB
    Free    : 4095 MB
    Unalloc: 3840 MB
    Memory :    8 MB          Session              Cummulative
                           ------------         -------------
 # Records Captured    :            594                41727
 # Records Published   :            792                18075

_ Filters: 1
_ Sources: 4
                                                                          Lag
  Targets: Conn #Records     Safe Restart         Last Commit     hhh:mm:ss.th
v -------- ---- -------- ------------------ ------------------- -------------
_ DB2APLY1 Yes       396       0001D0D12A81 2014-08-06 16:17:17   00:00:05.53
_ CSQ7/SQD Yes       396       0001D0D3C91B 2014-08-06 16:17:20   00:00:02.22
**************************** Bottom of data ********************************
```



IMS Data Capture Methods

Primary Methods of Capture

- ✓ Data Capture Exit Routines
- ✓ Log Based
- ✓ Application Based
- ✓ Hardware

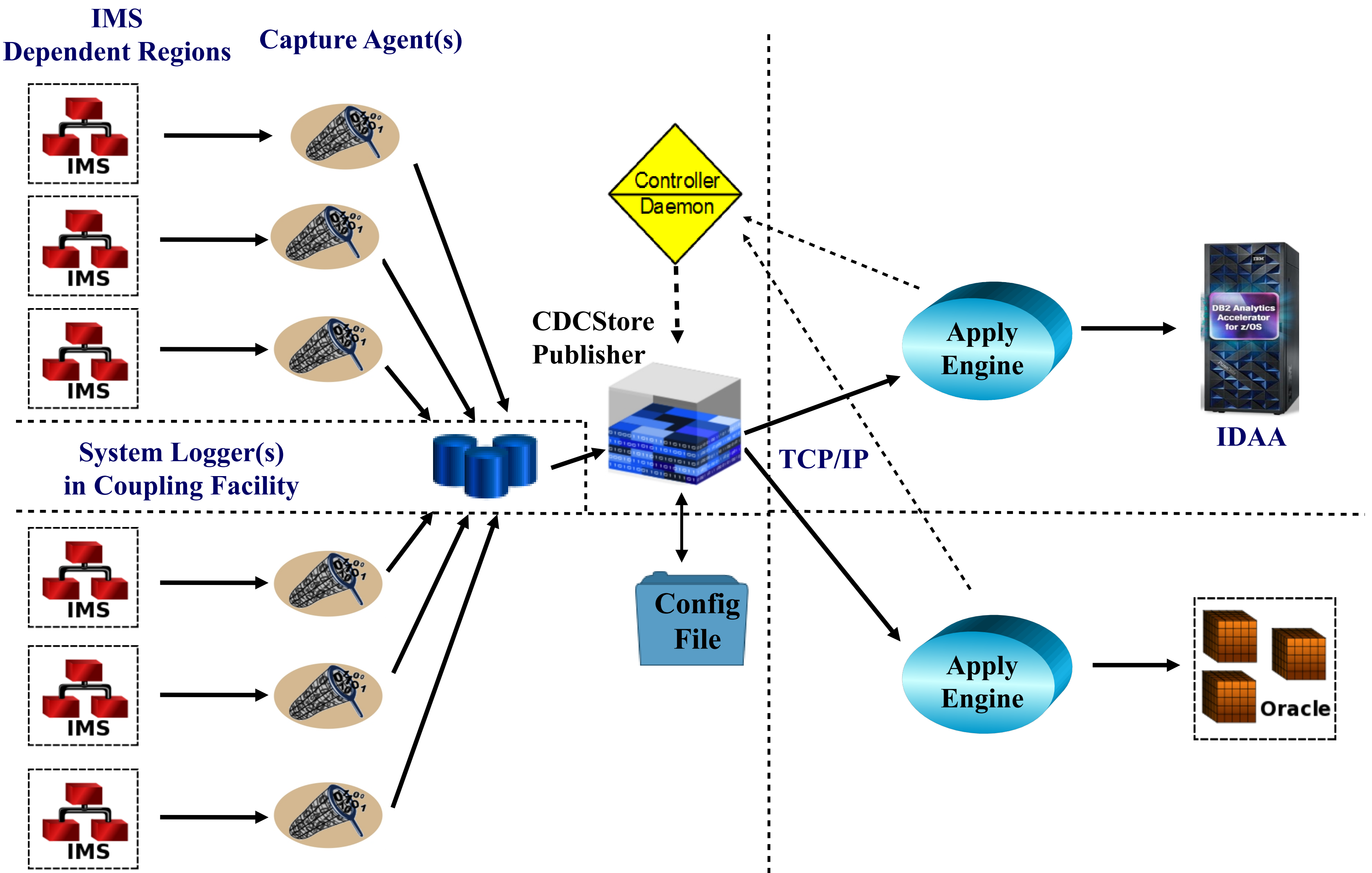
Database Exit Routines

- ✓ Near-Real-Time for IMS TM
- ✓ Scalability → Capture / Apply by FP Area, HALDB Partition, PSB, Database
- ✓ Do Not Require x'99' Log Records

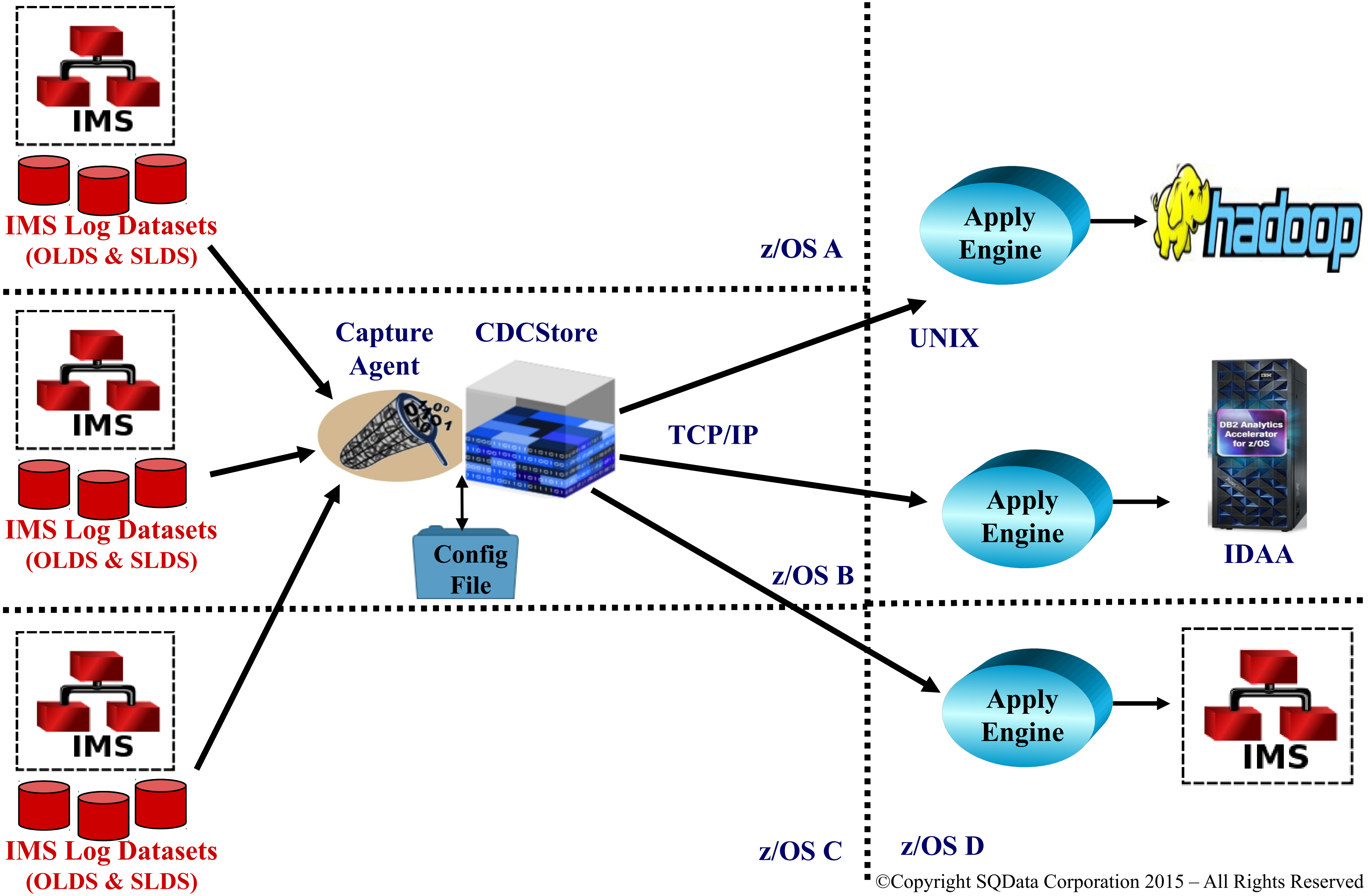
Log Based

- ✓ Near-Real-Time or Asynchronous
- ✓ Requires x'99' Log Records
- ✓ Allows for Recapture
- ✓ Scalability → Same as Database Exit Routines

SQData IMS zLog Replication

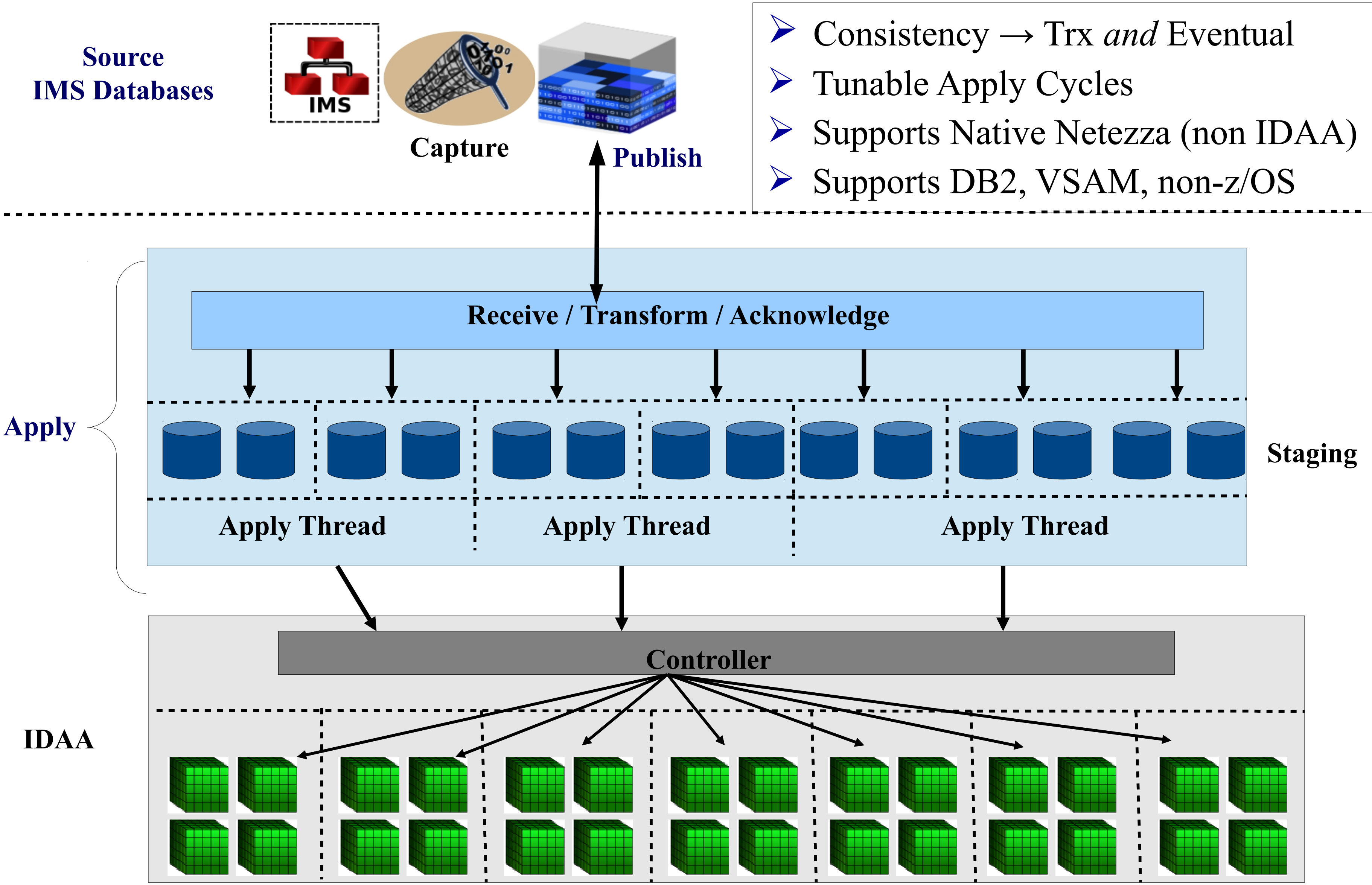


SQData IMS Log Replication



IMS to IDAA Replication Architecture

- Consistency → Trx *and* Eventual
- Tunable Apply Cycles
- Supports Native Netezza (non IDAA)
- Supports DB2, VSAM, non-z/OS



Agenda

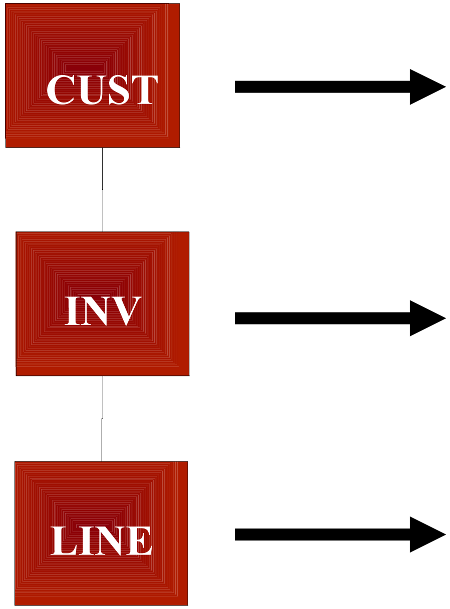
- Introduction
- IDAA Overview
- Replicating IMS to IDAA
- **Design Considerations**
- Q & A
- Conclusion

Common IMS Data Challenges

- **Code Page Translation**
- **Invalid Data**
  - ✓ Non-Numeric Data in Numeric Fields
  - ✓ Binary Zeros in Packed Fields (or Any Field)
  - ✓ Invalid Data in Character Fields
- **Dates**
  - ✓ Must be Decoded / Validated if Target Column is DATE or TIMESTAMP
  - ✓ May Require Knowledge of Y2K Implementation
  - ✓ Allow Extra Time for Date Intensive Applications
- **Repeating Groups**
  - ✓ Sparse Arrays
  - ✓ Number of Elements
  - ✓ Will Probably be De-normalized
- **Redefines**
- **Binary / 'Special' Fields**
  - ✓ Common in Older Applications Developed in 1970s / 80s
  - ✓ Generally Requires Application Specific Translation

Notes on Approach

- ✓ Each Segment Maps to One (1) or More Tables
- ✓ Helpful → Keep Source Fields and Target Column Names Similar
- ✓ Design Considerations
  - – Duration → Lower for Rehost...Higher for BI/DW
  - – Strong Target Data Types will Require Additional Transformation
  - – Be Careful to Avoid the 'Over Design'
- ✓ **<u>Best Practice</u>**: Keep Things as Simple as Possible



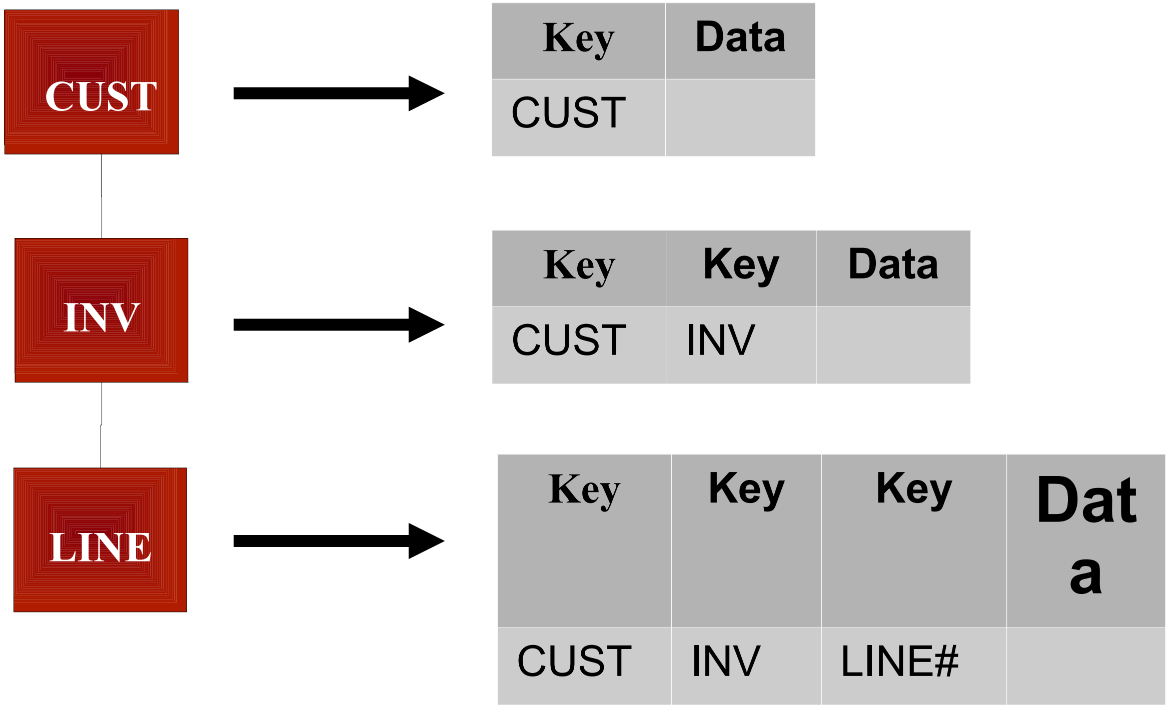
Key	Data
CUST	

Key	Key	Data
CUST	INV	

Key	Key	KeyA	KeyB	Data
CUST	INV	LINE-A	LINE-B	

IMS to Traditional Relational Model

- Normalized → at Least 2nd Normal Form
- Each Segment Typically Maps to One (1) or More Tables



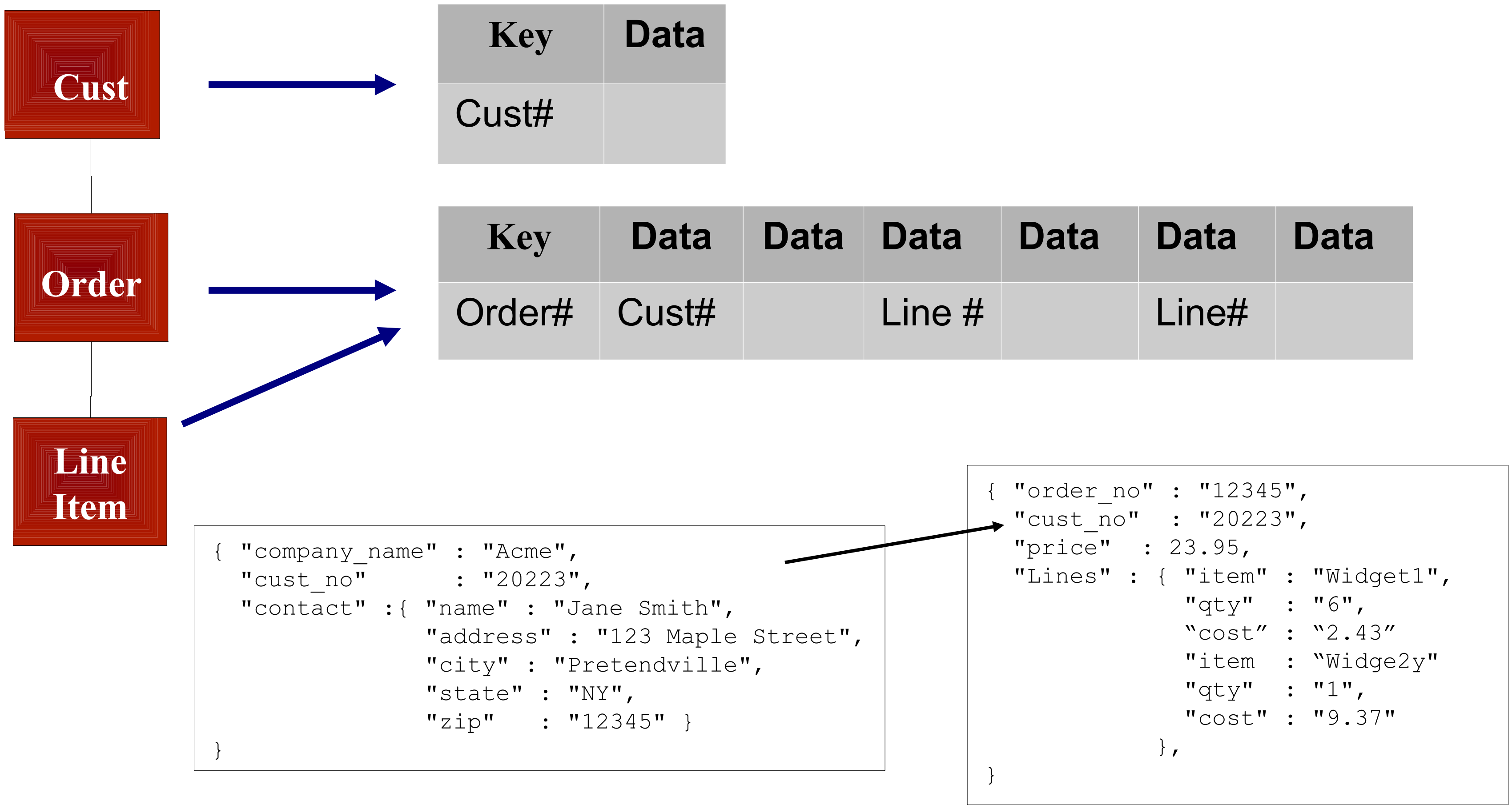
Key	Data
CUST	

Key	Key	Data
CUST	INV	

Key	Key	Key	Data
CUST	INV	LINE#	

Alternative → IMS to Big Data Model

- De-Normalized / Minimal Normalization
- Degree of Data Redundancy → Trade-Off for Query Performance



Key	Data
Cust#	

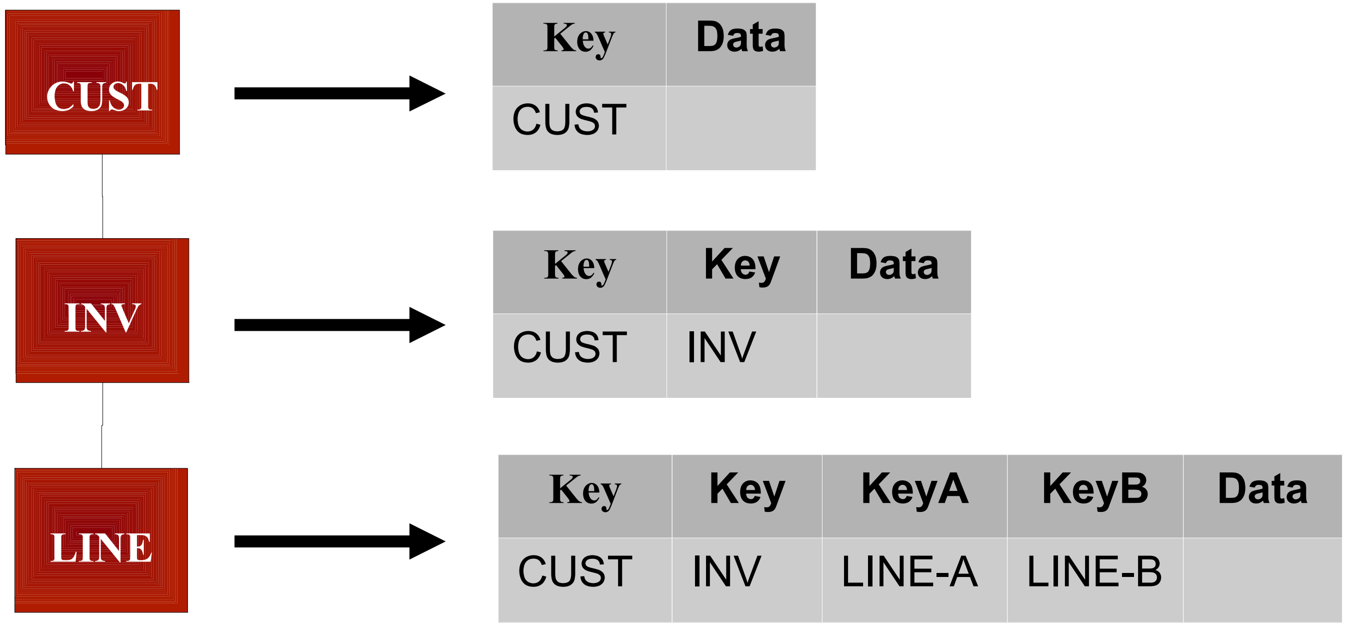
Key	Data	Data	Data	Data	Data	Data
Order#	Cust#		Line #		Line#	

```
{  "company_name"  :  "Acme",
   "cust_no"       :  "20223",
   "contact"  :{  "name"  :  "Jane Smith",
                  "address"  :  "123 Maple Street",
                  "city"  :  "Pretendville",
                  "state"  :  "NY",
                  "zip"    :  "12345"  }
}
```

```
{  "order_no"  :  "12345",
   "cust_no"   :  "20223",
   "price"   :  23.95,
   "Lines"  :  {  "item"  :  "Widget1",
                  "qty"   :  "6",
                  “cost”  :  “2.43”
                  "item   :  “Widge2y"
                  "qty"   :  "1",
                  "cost"  :  "9.37"
               },
}
```

Keys

- ✓ Fairly Straightforward → IMS Key Structure Simplifies Things
- ✓ Carry Parent Keys in Dependent Tables
- ✓ Use these Unique Keys as Distribution Keys in IDAA
- ✓ Plan on Keys being Comprised of Multiple Fields with Different Data Types
  - Character, Packed, Binary



CUST

INV

LINE

Key	Data
CUST	

Key	Key	Data
CUST	INV	

Key	Key	KeyA	KeyB	Data
CUST	INV	LINE-A	LINE-B	

Redefined Fields

- ✓ Extends Analysis Timeline More Often than Not
- ✓ Requires Consult with SME and/or Research to Determine Which Field to Use
- ✓ Options for Simple Redefines:
  - Map Least Restrictive Field (PIC X)
  - Map Both Fields

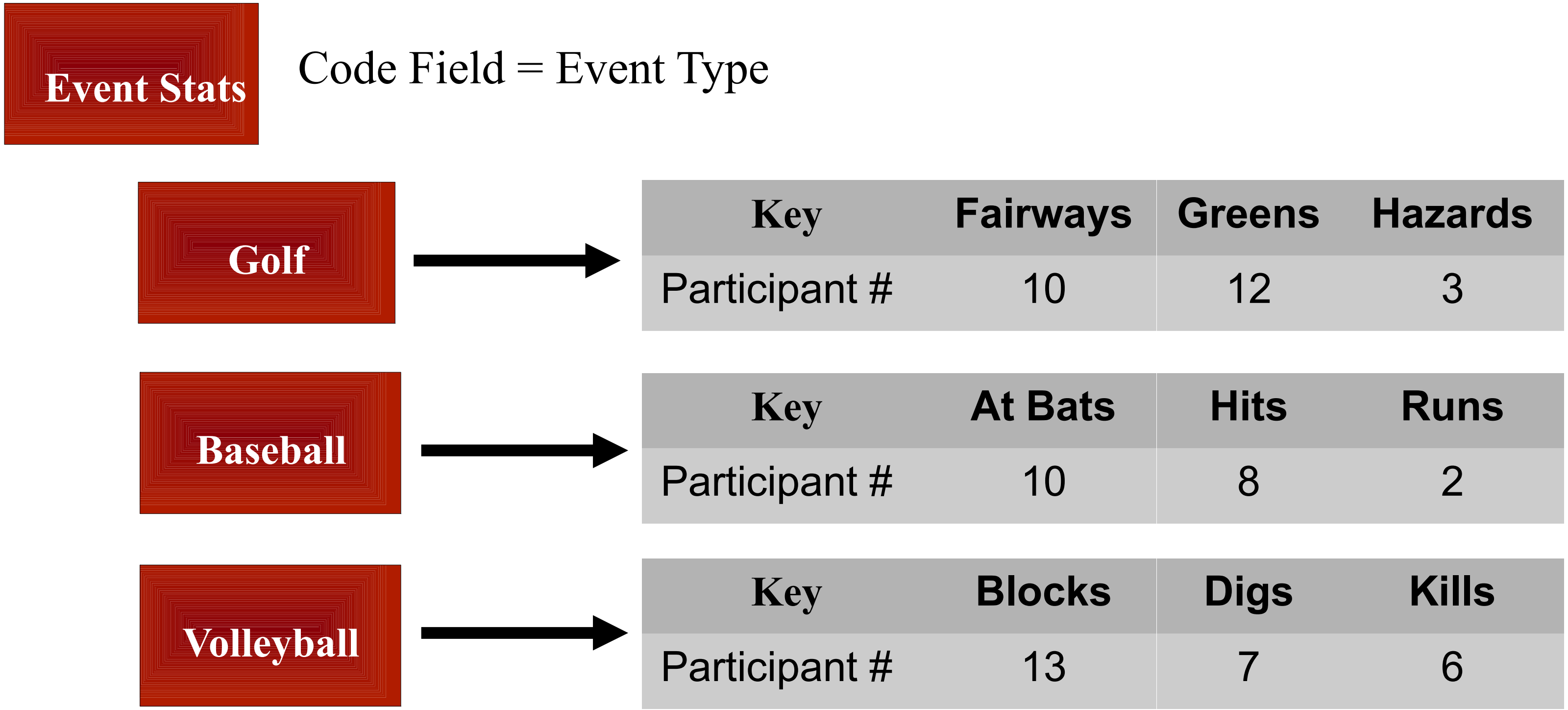
```
05   ACCOUNT-ID                                          PIC 9(7).
05   ACCOUNT-ID REDEFINES ACCOUNT-NO                     PIC X(7).
```

- ✓ Options for Complex Redefines:
  - Map More Granular Field(s) → Will Require More Data Cleansing / Transformation
  - Map All Fields

```
05   ACCOUNT-ID                                      PIC X(5).
05   ACCOUNT-ID REDEFINES ACCOUNT-NO.
       10  ACCOUNT-PREFIX                   PIC X(1).
       10 ACCOUNT-NUMBER                    PIC S9(7) COMP-3.
```

Redefined Segments: Full

- ✓ Redefine Generally Identified by One (1) or More Code Fields
- ✓ Each Redefine Mapped to a Separate Target Table



Event Stats

Code Field = Event Type

Golf

Key	Fairways	Greens	Hazards
Participant #	10	12	3

Baseball

Key	At Bats	Hits	Runs
Participant #	10	8	2

Volleyball

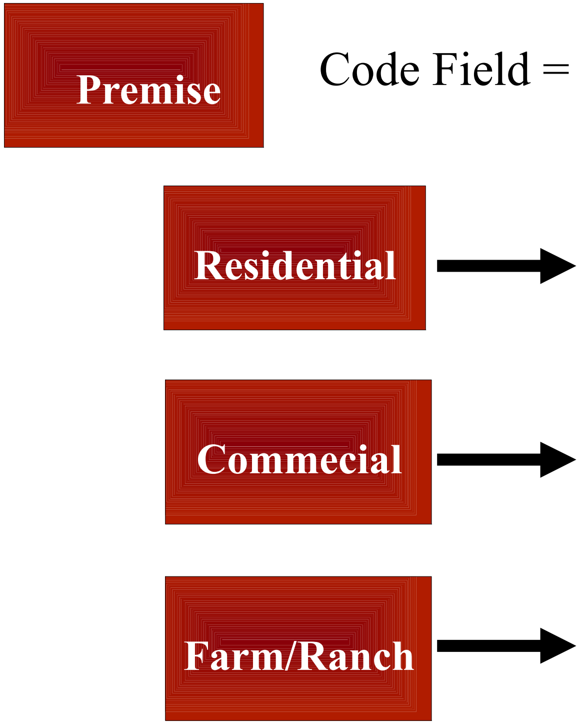
Key	Blocks	Digs	Kills
Participant #	13	7	6

Redefined Segments: Partial

- ✓ Redefine Generally Identified by One (1) or More Code Fields
- ✓ Redefines can be Mapped to the Same Target Table if Enough Fields in Common

or

- ✓ Each Redefine Mapped to a Separate Target Table



Premise

Code Field = Premise Type

Residential

Key1	Key2	Addr	Pool Size	Tenants	Crop
PR#	PR_Type	123 Elm	25,000	null	null

Commecial

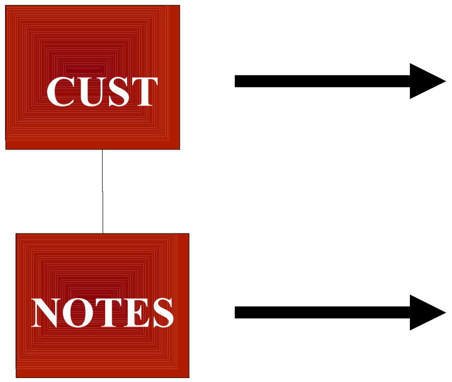
Key1	Key2	Addr	Pool Size	Tenants	Crop
PR#	PR_Type	456 Ash	null	38	null

Farm/Ranch

Key1	Key2	Addr	Pool Size	Tenants	Crop
PR#	PR_Type	456 Ash	null	null	Corn

Non-Keyed Segments

- ✓ Commonly Used for Text / Comments
- ✓ Straightforward for ETL
  - – Unload in Order of Occurrence
  - – Optional: Use a Sequence Number to Keep Things in Order on Target Side
- ✓ Tricky for CDC
  - – Only Have Access to Parent Key(s)
  - – Option 1: Set Apply Key to Include All Non-Keyed Data (exclude sequence #)
  - – Option 2: Fully Materialize All Non-Keyed Segments when 1 Changes
  - – Make Sure Your ETL/CDC Tool Can Handle Non-Keyed Segments

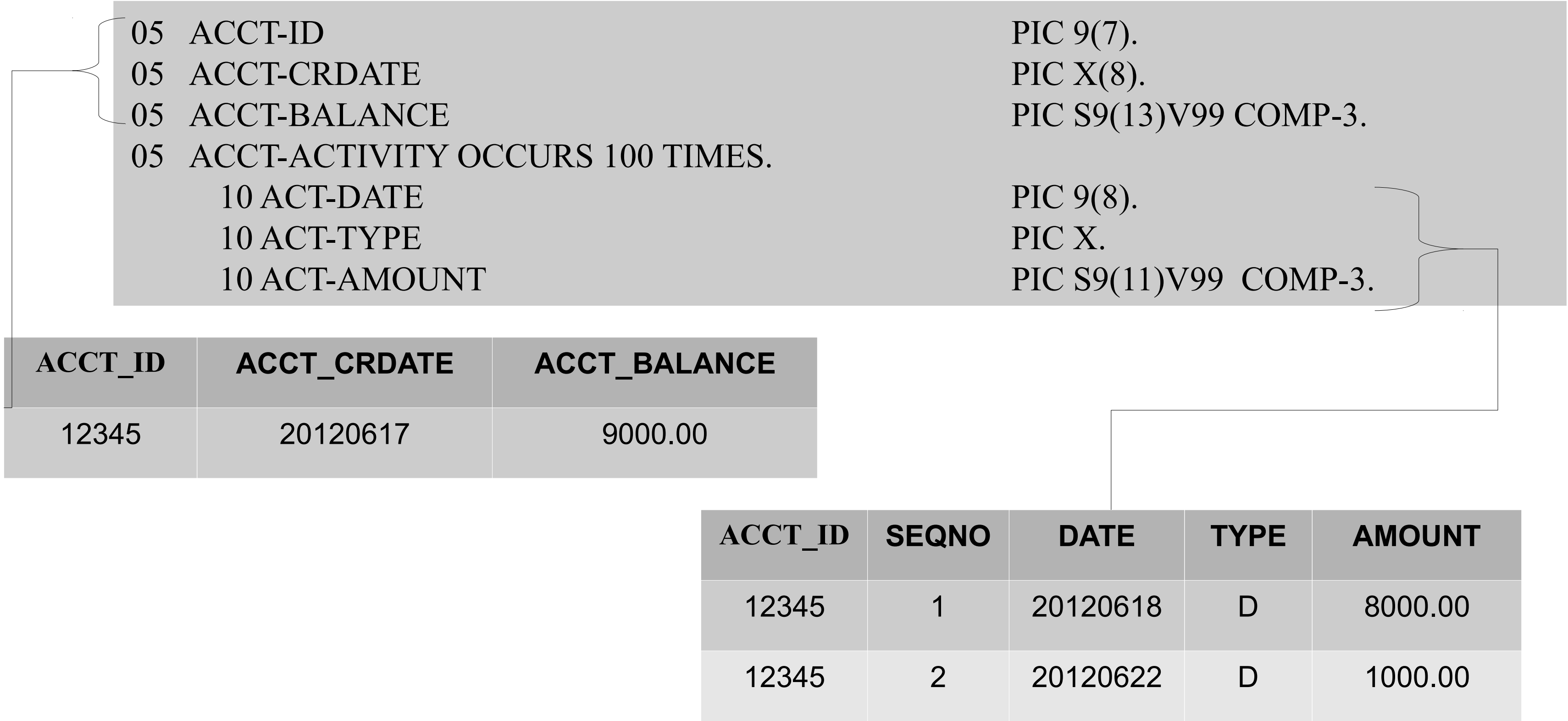


Key	Data
C123	

Key	SEQNO	Data
C123	1	abcdefghij
C123	2	klmnopqrs

Repeating Groups: Relational

- ✓ Typical Candidates for Normalization Based on # Occurs
- ✓ Options:
  - Low # Occurs → Keep in Same Table as Rest of Segment
  - Map to Separate Table – Requires a Sequence Number



```
05   ACCT-ID                                             PIC 9(7).
05   ACCT-CRDATE                                         PIC X(8).
05   ACCT-BALANCE                                        PIC S9(13)V99 COMP-3.
05   ACCT-ACTIVITY OCCURS 100 TIMES.
        10 ACT-DATE                                      PIC 9(8).
        10 ACT-TYPE                                      PIC X.
        10 ACT-AMOUNT                                    PIC S9(11)V99  COMP-3.
```

ACCT_ID	ACCT_CRDATE	ACCT_BALANCE
12345	20120617	9000.00

ACCT_ID	SEQNO	DATE	TYPE	AMOUNT
12345	1	20120618	D	8000.00
12345	2	20120622	D	1000.00

Alternative → Repeating Groups: Big Data

- ✓ All Occurrences into the Same Target
- ✓ No Need for Sequence Number

```
05   ACCT-ID                                                PIC 9(7).
05   ACCT-CRDATE                                            PIC X(8).
05   ACCT-BALANCE                                           PIC S9(13)V99 COMP-3.
05   ACCT-ACTIVITY OCCURS 100 TIMES.
        10 ACT-DATE                                         PIC 9(8).
        10 ACT-TYPE                                         PIC X.
        10 ACT-AMOUNT                                       PIC S9(11)V99  COMP-3.
```

ACCT_ID	ACCT_CRDATE	BALANCE
12345	20120617	9000.00

DATE	TYPE	AMOUNT	DATE	TYPE	AMOUNT
20120618	D	8000.00	20120622	D	1000.00

Summary

- Let the Business Drive the Effort
- Temper the Exuberance
- Align with Enterprise Data
- Lose the Relational Model Mentality
- Use an Iterative Approach for Implementation
- Select the Correct Tool Vendor
  - ✓ Specializes in Heterogeneous Data Movement
  - ✓ Bulk Data Extract & Changed Data Capture / Replication
  - ✓ Competent to Participate with Design & Deployment

Where to Find Additional Information

- Email Requests
  - North America: info@sqdata.com
  - EMEA: info@insoft-software.de

- Phone Requests
  - North America: 866-252-3575
  - EMEA: Antoine Gruson – +33 (0)5 34 36 29 22

- Websites
  - SQData: www.sqdata.com
  - Insoft: www.insoft-software.de

SQDATA CORPORATION

IMS to IDAA

Keeping Your Data Current

Prepared for the:

IMS Tech Symposium

17 March 2015