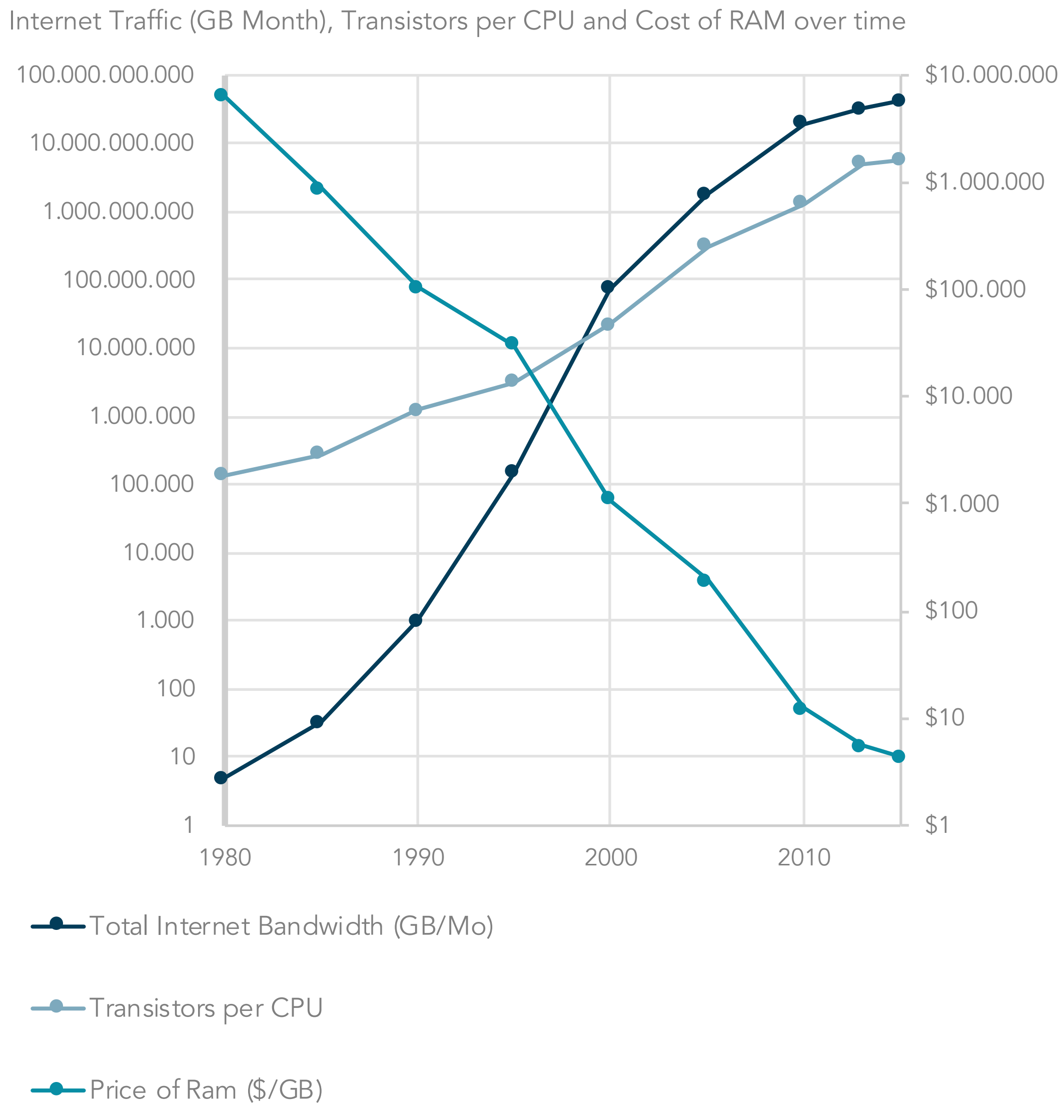
VoltDB

David Rolfe
Director of Solution Architecture, EMEA

Doug Jauregui
Sr. Solutions Engineer

Legacy database technology is obsolete



- Legacy RDBMS designs date from about 1985.
- Vendors are finding legacy databases increasingly uneconomic.
- Legacy databases struggle to scale beyond 2 nodes.
- But demand for transactions is increasing all the time.
- Meanwhile Moore's law means hardware and RAM keeps getting cheaper

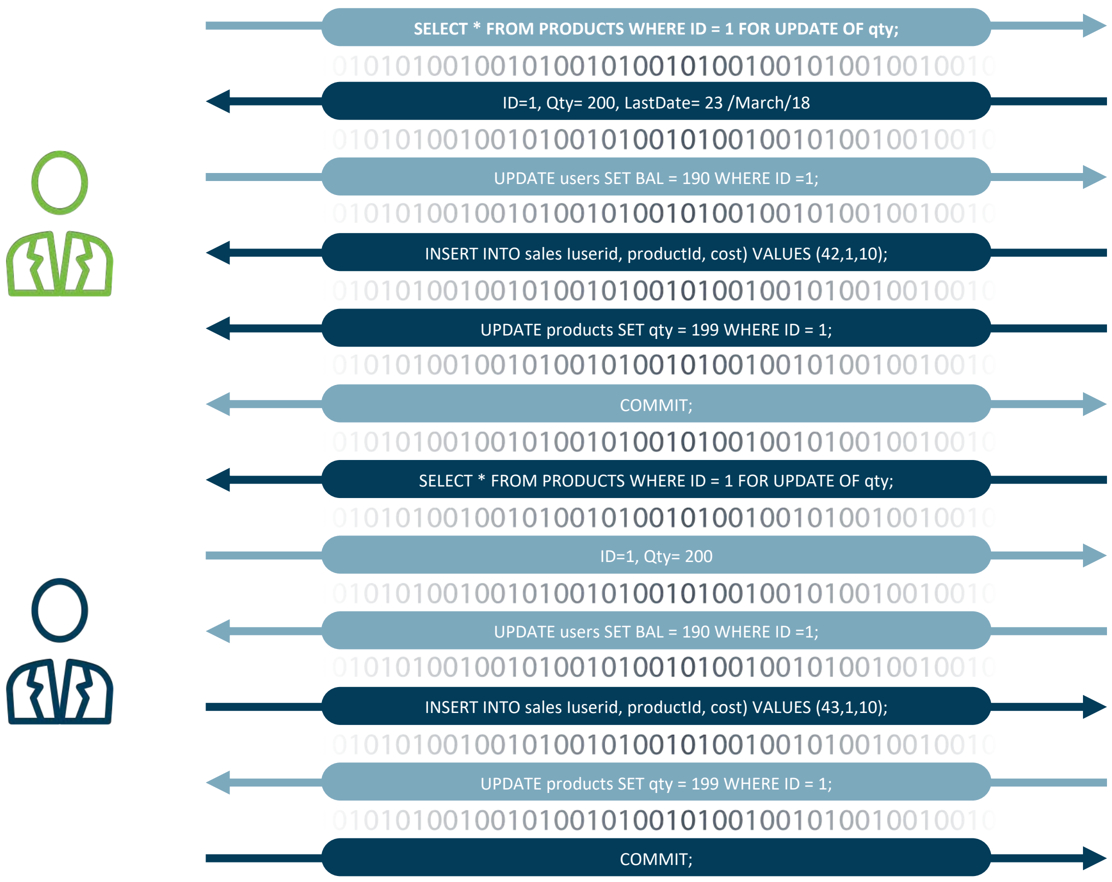
VOLTDB

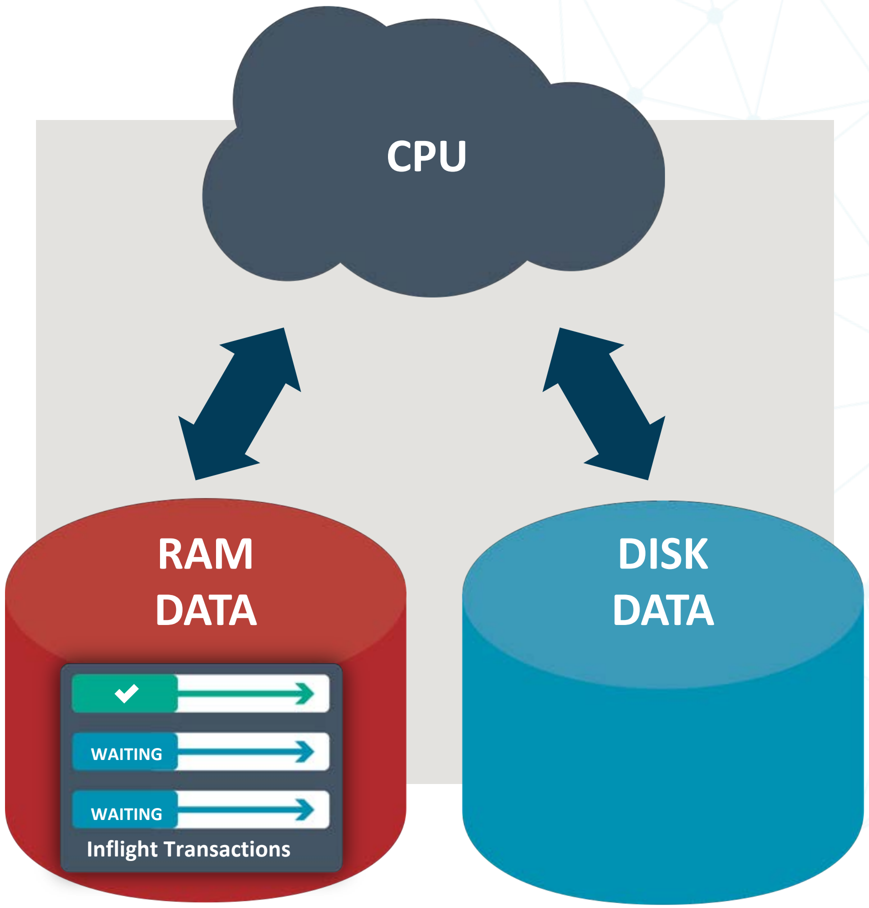
21^{st} Century Requirements for transaction processing

- Virtualization friendly .
- ACID transactions.
- Millisecond response times.
- No ”Long Tail”
- Supports complicated logic
- Easily scalable beyond 2 nodes.
- HA “Just Happens”
- Geo replication
- “Translytics”/”HTAP”

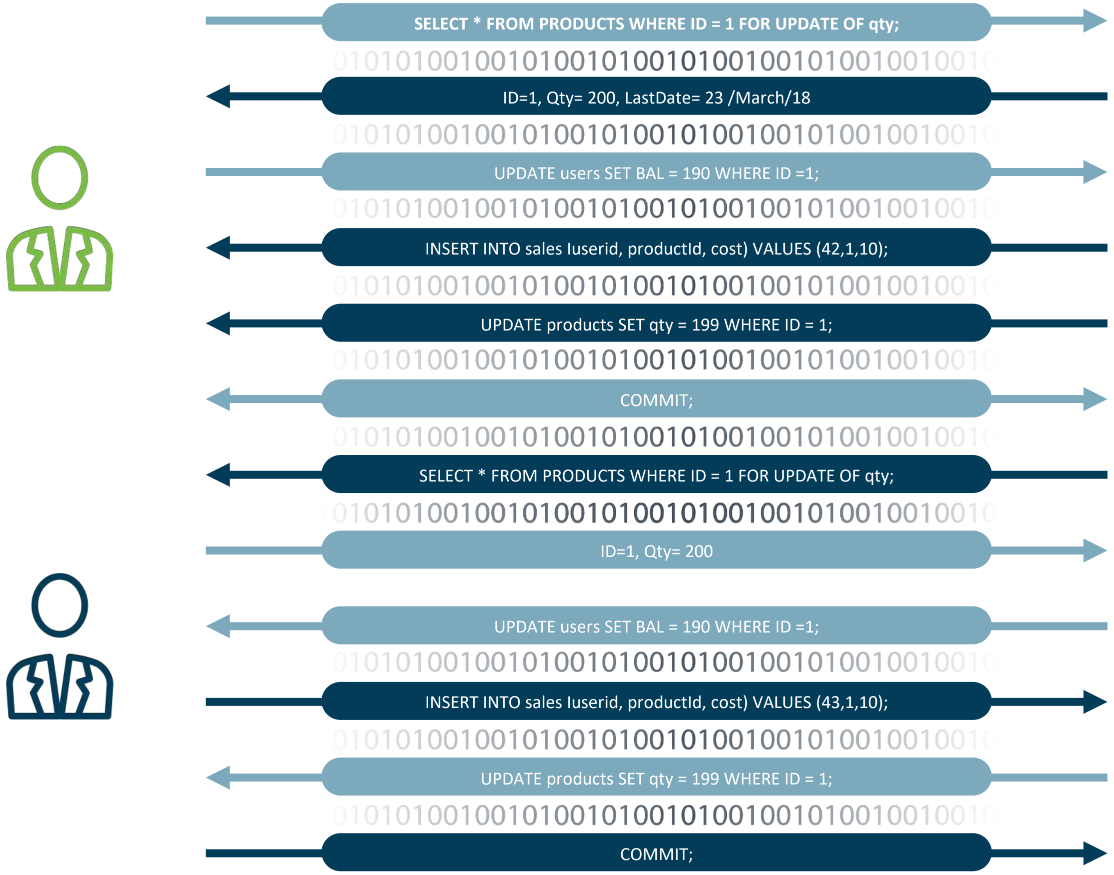
VOLTDB

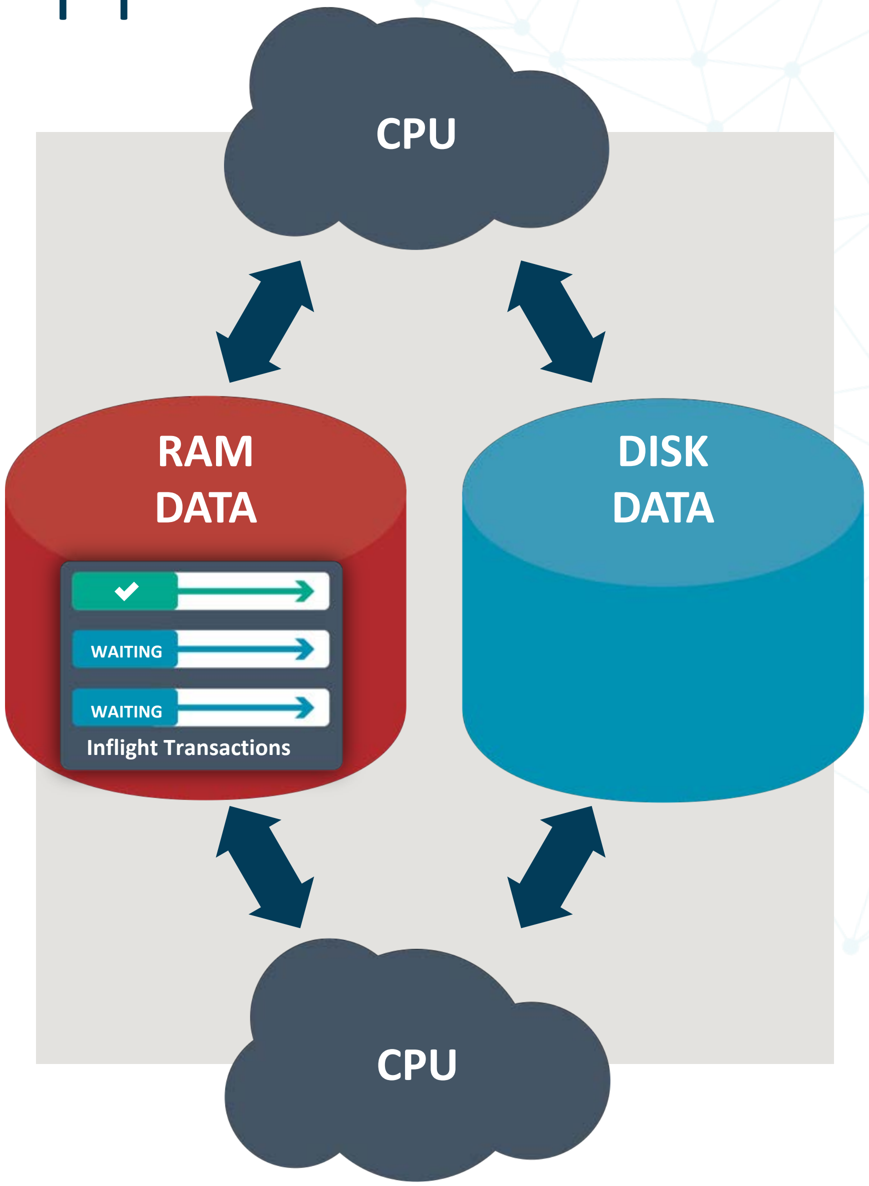
RDBMS - How We Thought an RDBMS Worked



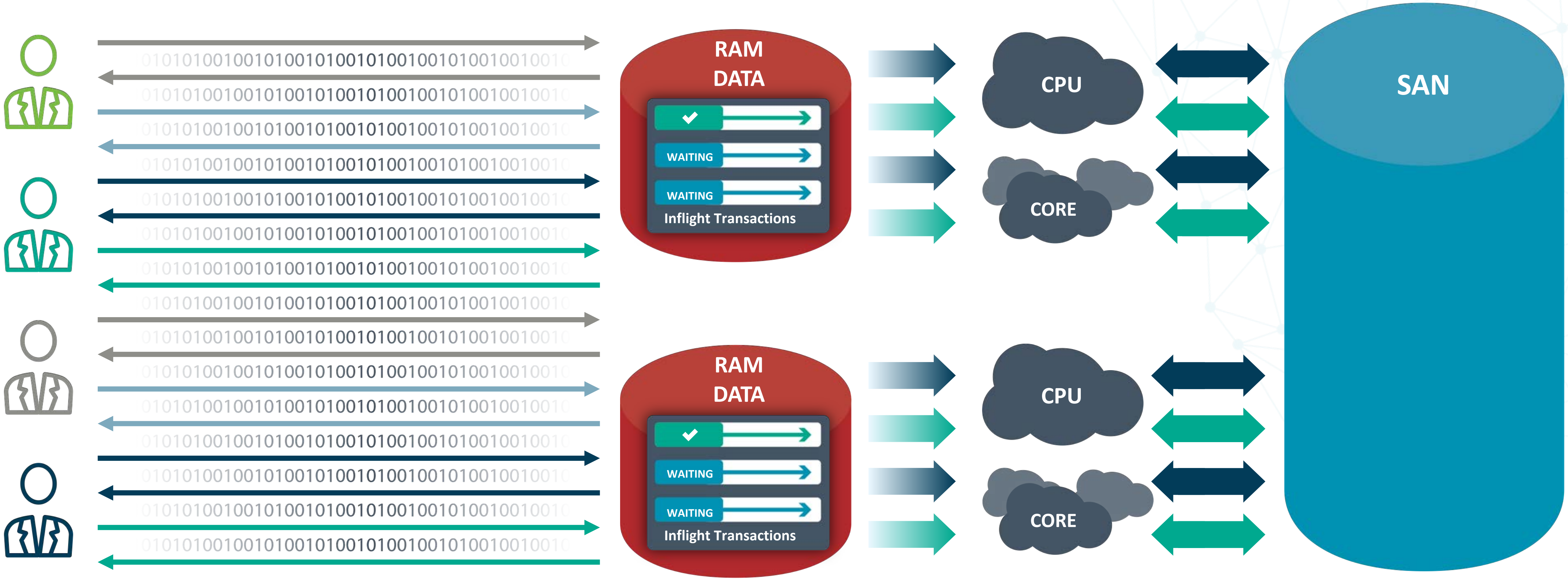


RDBMS - What Actually Happens – Part 1…





RDBMS - What Actually Happens – Part 2

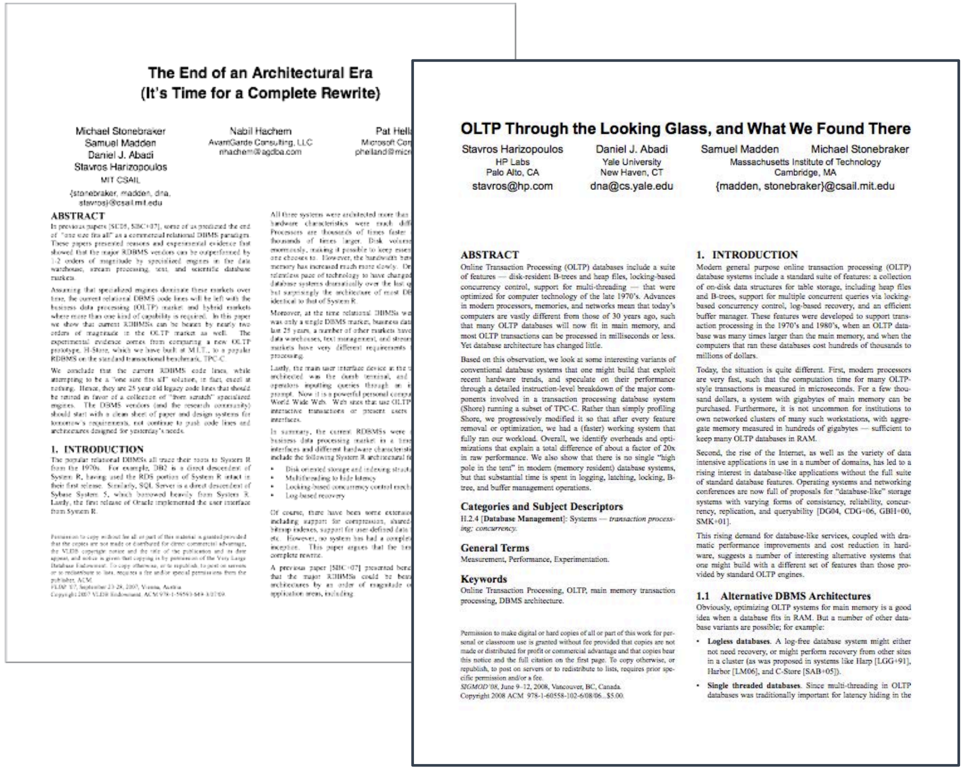


VOLTDB

If we tried this in a supermarket…

VOLTDB

Dr. Michael Stonebraker found a solution..



The End of an Architectural Era
(It's Time for a Complete Rewrite)

Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos — MIT CSAIL
Nabil Hachem — AvantGarde Consulting, LLC
Pat Helland — Microsoft Cor...

ABSTRACT

1. INTRODUCTION

OLTP Through the Looking Glass, and What We Found There

Stavros Harizopoulos — HP Labs, Palo Alto, CA — stavros@hp.com
Daniel J. Abadi — Yale University, New Haven, CT — dna@cs.yale.edu
Samuel Madden, Michael Stonebraker — Massachusetts Institute of Technology, Cambridge, MA — {madden, stonebraker}@csail.mit.edu

ABSTRACT

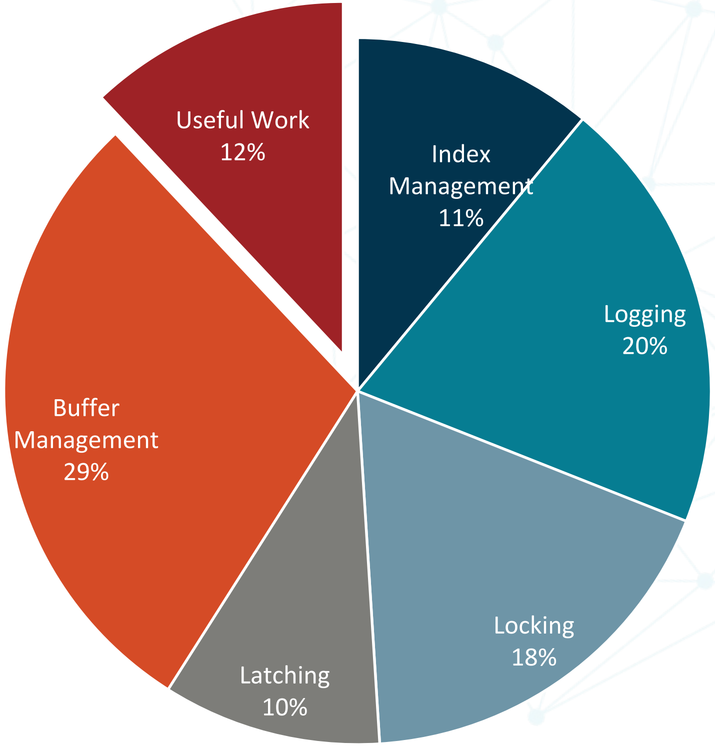
Categories and Subject Descriptors

General Terms

Keywords

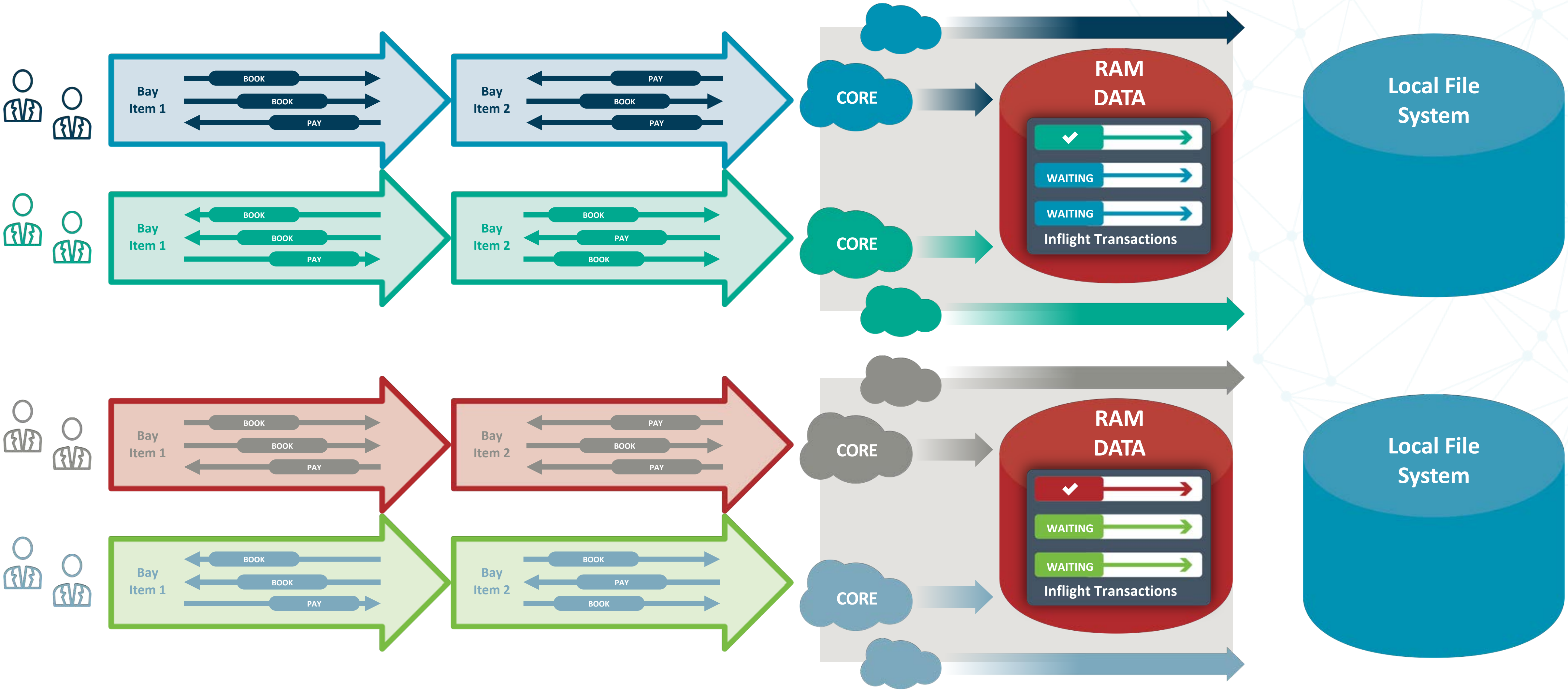
1. INTRODUCTION

1.1 Alternative DBMS Architectures



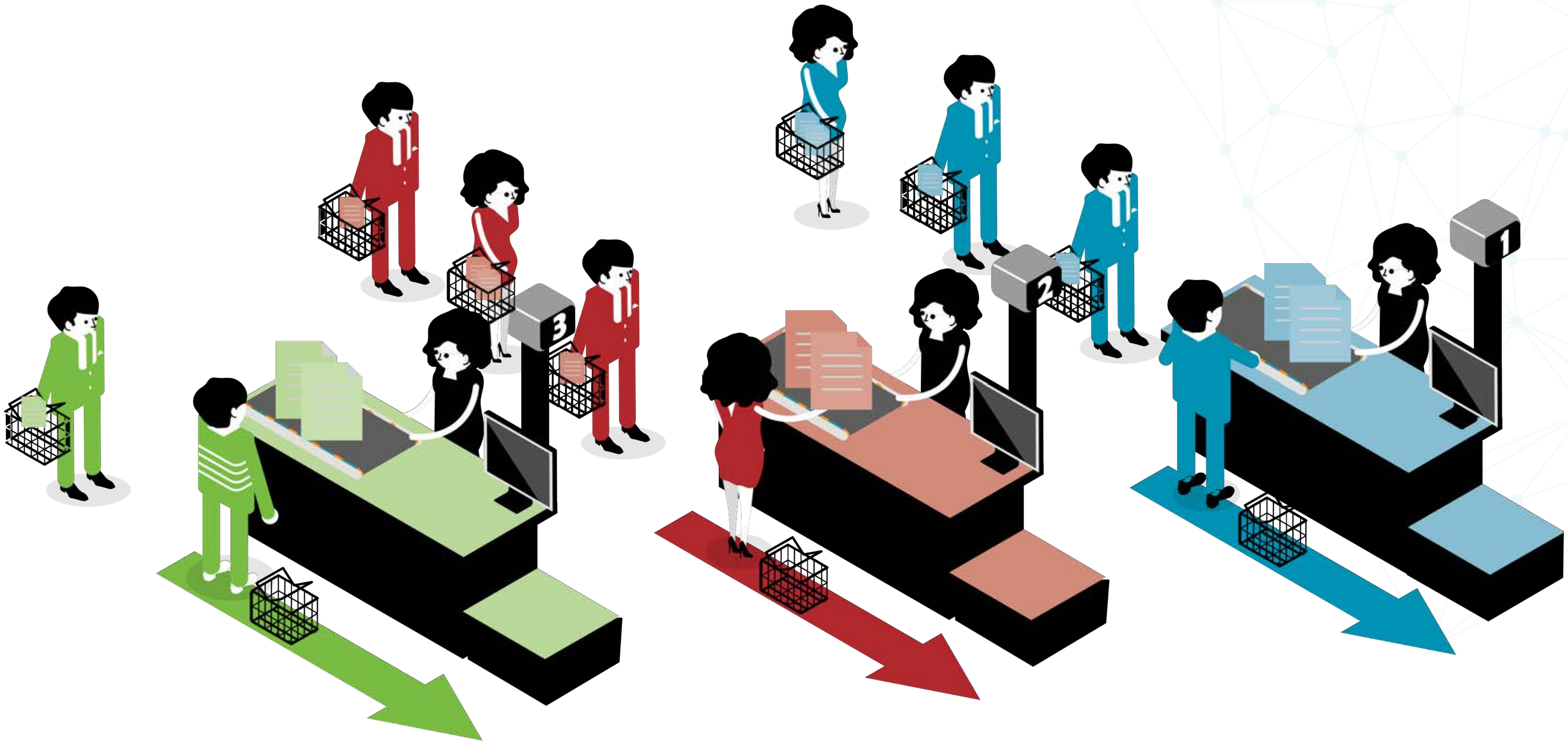
VOLTDB

How VoltDB works

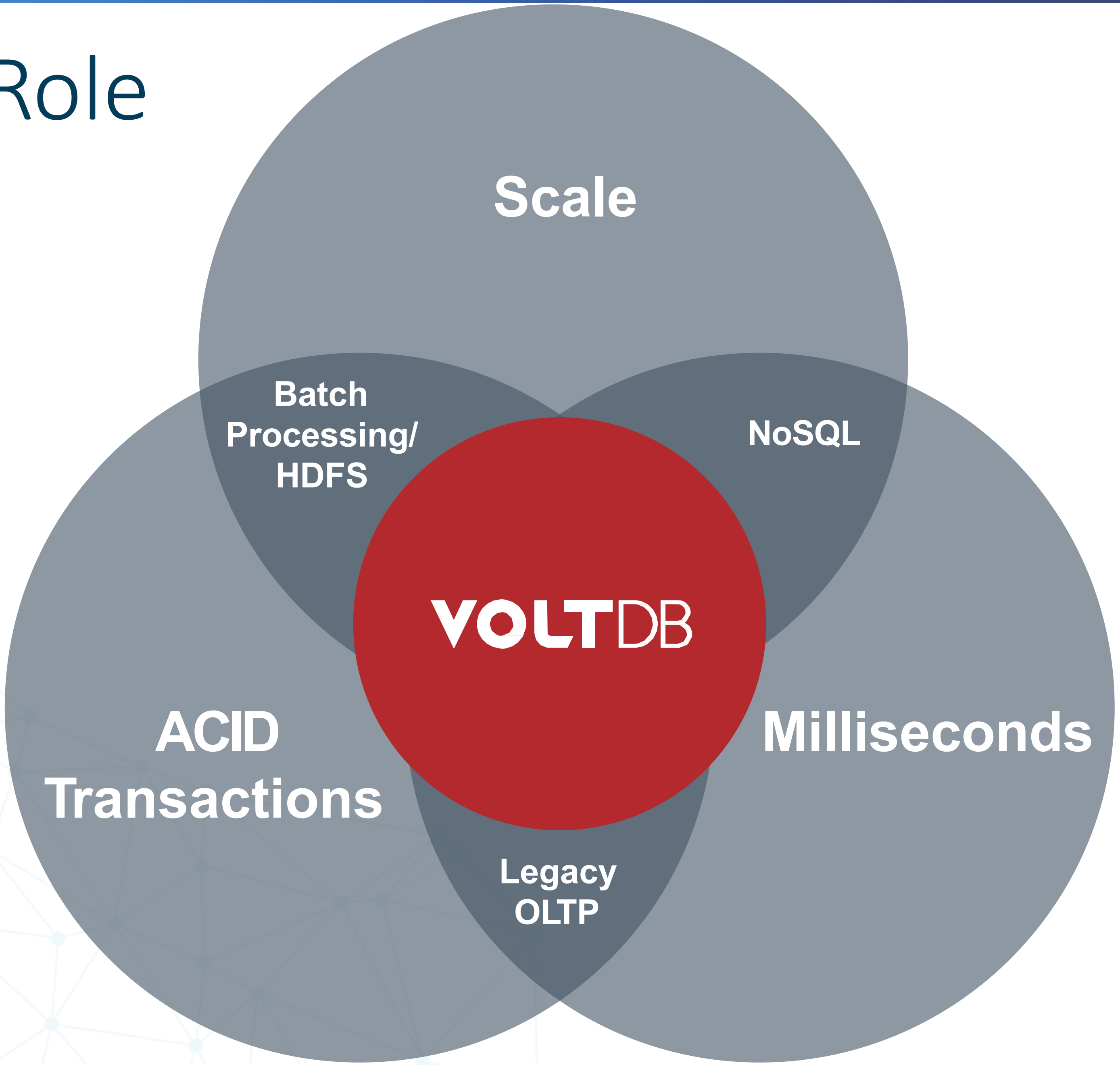


VOLTDB

How a supermarket works...



VoltDB's Role



The only 3 ways to interact with any database

Approach	Examples	Strengths	Weaknesses
Many SQL Statements + Commit or Rollback	JDBC, ODBC,	Liked by developers, initial development is rapid	• Doesn't handle scaling OLTP loads well – DB spends its time figuring out who can see what instead of working • Constant locking problems for shared, finite resources • Failure of a client to Commit or Rollback causes a temporary resource leak
Move all the data to the client and back again	NoSQL, KV Stores	Very developer friendly	• Multiple updated copies of the data can arrive at the same time for scaling OLTP loads • All of the data gets moved across the network, every time.
Stored Procedures	VoltDB, PL/SQL	Predictable speed and best possible scaling characteristics	• Not in fashion with developers. • PL/SQL created perception of complexity. • Other implementations of Java Stored Procedures really slow.

VOLTDB

A Proven and Reliable Partner

Telco

Billing/rights management, subscriber data, etc.

Financial Services

Risk, market data management, customer mgt.

Personalize, Customize, Target

Ad optimization, audience segmenting, customer service

IoT Platforms, Energy, Sensor

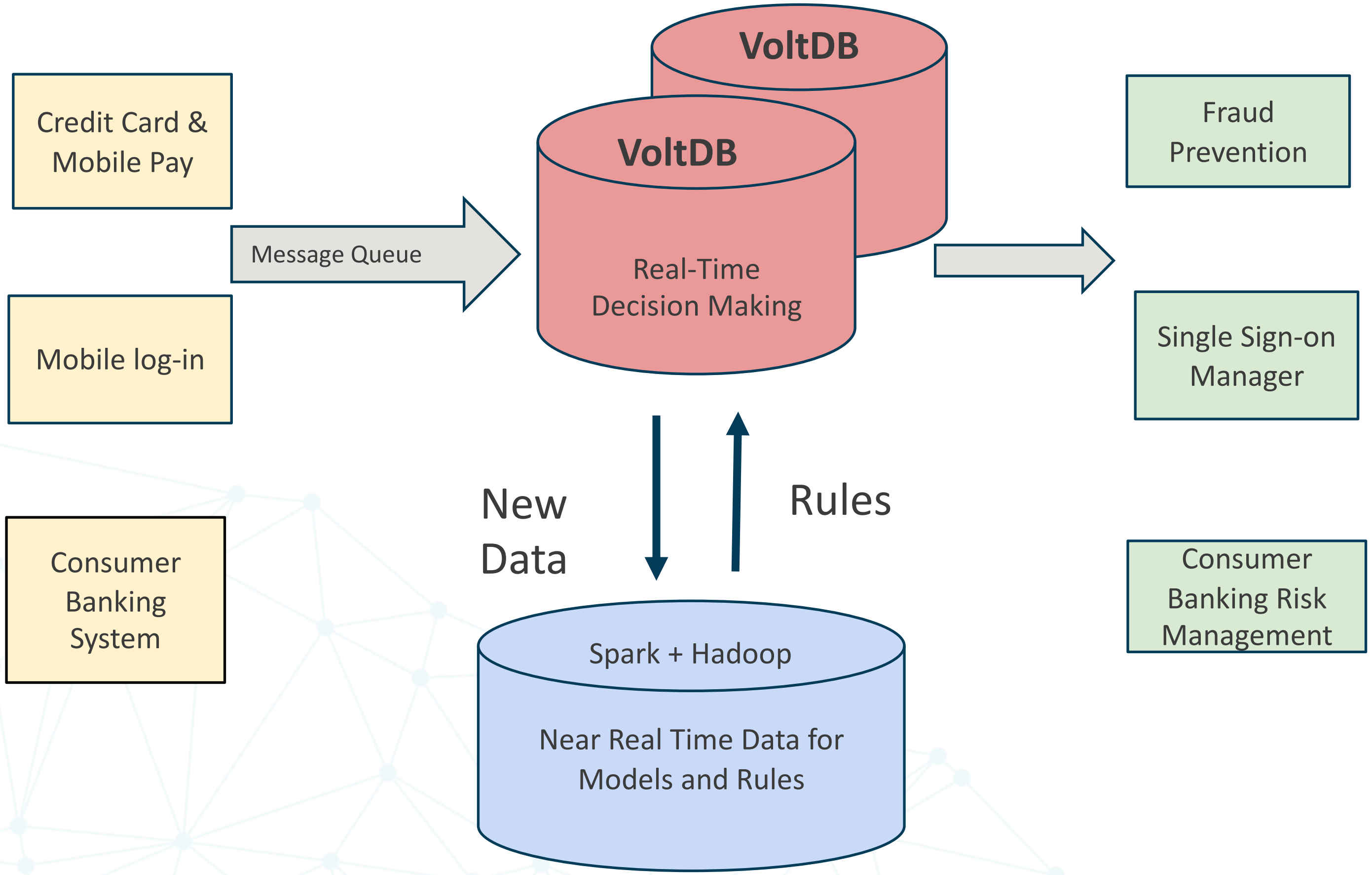
Smart grid/meters, asset tracing & management

Infrastructure, Dashboards, KPIs

Data pipeline, system performance, streaming ETL.

VoltDB & Machine Learning

HUAWEI



Application/Use Case

- Fraud Prevention
- Single sign-in of all Huawei phones
- Consumer banking risk management

Why VoltDB?

- > 50% reduction in fraud cases
- > $15M/year saved from fraud loss
- 10k complex Transactions Per Second
- 99.99% transactions finish < 50ms
- 10x better performance than traditional fraud detection

VOLTDB

VoltDB & Machine Learning

- VoltDB has a C++ core with a Java layer on top for running stored procedures
- VoltDB implements High Availability by running the same code in two places at once.
- Any Java class can be used in a stored procedure call provided:
  - It's deterministic (all copies of the code have to act the same way…)
  - It doesn't access network resources (which would make it non-deterministic)
- Examples: H20.AI and (J)PMML

ML Example – User Defined Function in H20

```java
public class AirlineDemoUDF {

    private static String modelClassName = "gbm_pojo_test";

    public String ademo(String cRSDepTime, String year, String month, String dayOfMonth, String dayOfWeek,
            String uniqueCarrier, String origin, String dest) {

        try {

            hex.genmodel.GenModel rawModel;
            rawModel = (hex.genmodel.GenModel) Class.forName(modelClassName).newInstance();
            EasyPredictModelWrapper model = new EasyPredictModelWrapper(rawModel);

            RowData row = new RowData();
            row.put("Year", year);
            row.put("Month", month);
            row.put("DayofMonth", dayOfMonth);
            row.put("DayOfWeek", dayOfWeek);
            row.put("CRSDepTime", cRSDepTime);
            row.put("UniqueCarrier", uniqueCarrier);
            row.put("Origin", origin);
            row.put("Dest", dest);
            BinomialModelPrediction p = model.predictBinomial(row);

            return (p.label);

        } catch (Exception e) {

            System.err.println(e.getMessage());
            return null;

        }

    }
```

```sql
CREATE FUNCTION ademo FROM METHOD h20.AirlineDemoUDF.ademo;

CREATE PROCEDURE flight_hist
PARTITION ON TABLE   flights COLUMN  f_FlightNum AS
SELECT f_cRSDepTime,  f_year,  f_month,  f_dayOfMonth,
f_dayOfWeek, f_uniqueCarrier,  f_origin,  f_dest
,ademo(f_cRSDepTime,  f_year,  f_month,  f_dayOfMonth,
f_dayOfWeek, f_uniqueCarrier,  f_origin,  f_dest ) ademo
from flights
where f_FlightNum = ?
order by f_year,  f_month,  f_dayOfMonth,f_cRSDepTime;
```

VOLTDB

ML Example – Calling JPMML from a Procedure

```java
public VoltTable[] runModel(String pmmlFileName, VoltTable inputParams) throws Exception {

    Evaluator evaluator = pmmlEvaluators.get(pmmlFileName);

    if (evaluator == null) {
        throw new Exception("Model " + pmmlFileName + " not found");
    }

    List<InputField> inputFields = evaluator.getInputFields();
    Map<FieldName, FieldValue> arguments = new LinkedHashMap<FieldName, FieldValue>();

    // Sanity check input params

    if (inputParams == null) {
        throw new Exception("VoltTable inputParams can't be null");
    }

    if (inputParams.getRowCount() != 1) {
        throw new Exception("VoltTable inputParams must have one row");
    }

    if (inputParams.getColumnCount() != inputFields.size()) {
        throw new Exception("VoltTable inputParams must match length of inputFields. inputParams
                + inputParams.getColumnCount() + " columns, expect " + inputFields.size());
    }

    inputParams.advanceRow();
    for (InputField inputField : inputFields) {
        mapVoltparamToPmmlParam(inputParams, arguments, inputField);
    }

    Map<FieldName, ?> result = evaluator.evaluate(arguments);

    // Processing results
    // Retrieving the values of target fields (ie. primary results):
    List<TargetField> targetFields = evaluator.getTargetFields();
    VoltTable resultTable = mapPmmlTargetFieldsToVoltTable(result, targetFields);

    // other fields
    List<OutputField> outputFields = evaluator.getOutputFields();
    VoltTable otherTable = mapPmmlOutputFieldsToVoltTable(result, outputFields);

    VoltTable[] outputParams = { resultTable, otherTable };

    return outputParams;

}
```

```java
public class GolfDemo extends VoltProcedure {

    public VoltTable[] run(double temperature, double humidity,
            String windy, String outlook) throws VoltAbortException {

        VoltTable[] pmmlOut;

        try {

            JPMMLImpl i = JPMMLImpl.getInstance();
            VoltDBJPMMLWrangler w = i.getPool().borrowObject();
            final String modelName = "tree.model";
            VoltTable paramtable = w.getEmptyTable(modelName);
            paramtable.addRow(temperature, humidity, windy, outlook);
            pmmlOut = w.runModel(modelName, paramtable);

        } catch (Exception e) {

            System.err.println(e.getMessage());
            throw new VoltAbortException(e);

        }

        voltExecuteSQL(true);
        return pmmlOut;
    }

}
```

VOLTDB

For more information:

www.voltdb.com

drolfe@voltdb.com

VOLTDB