

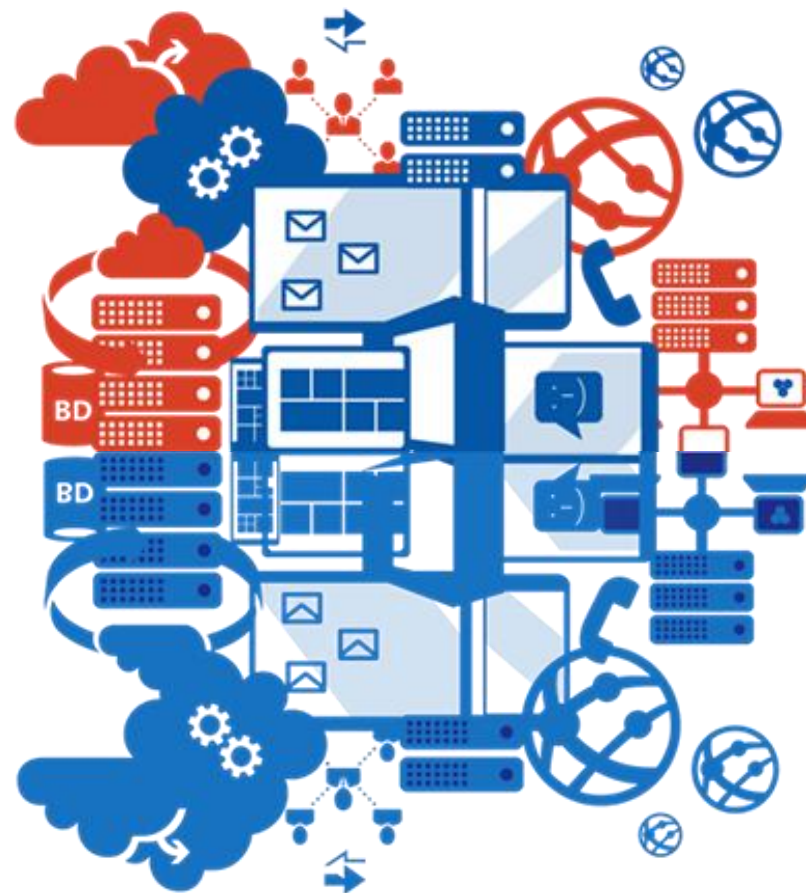


SQL Server 2016

New innovations

César Mendes
cmendes@microsoft.com

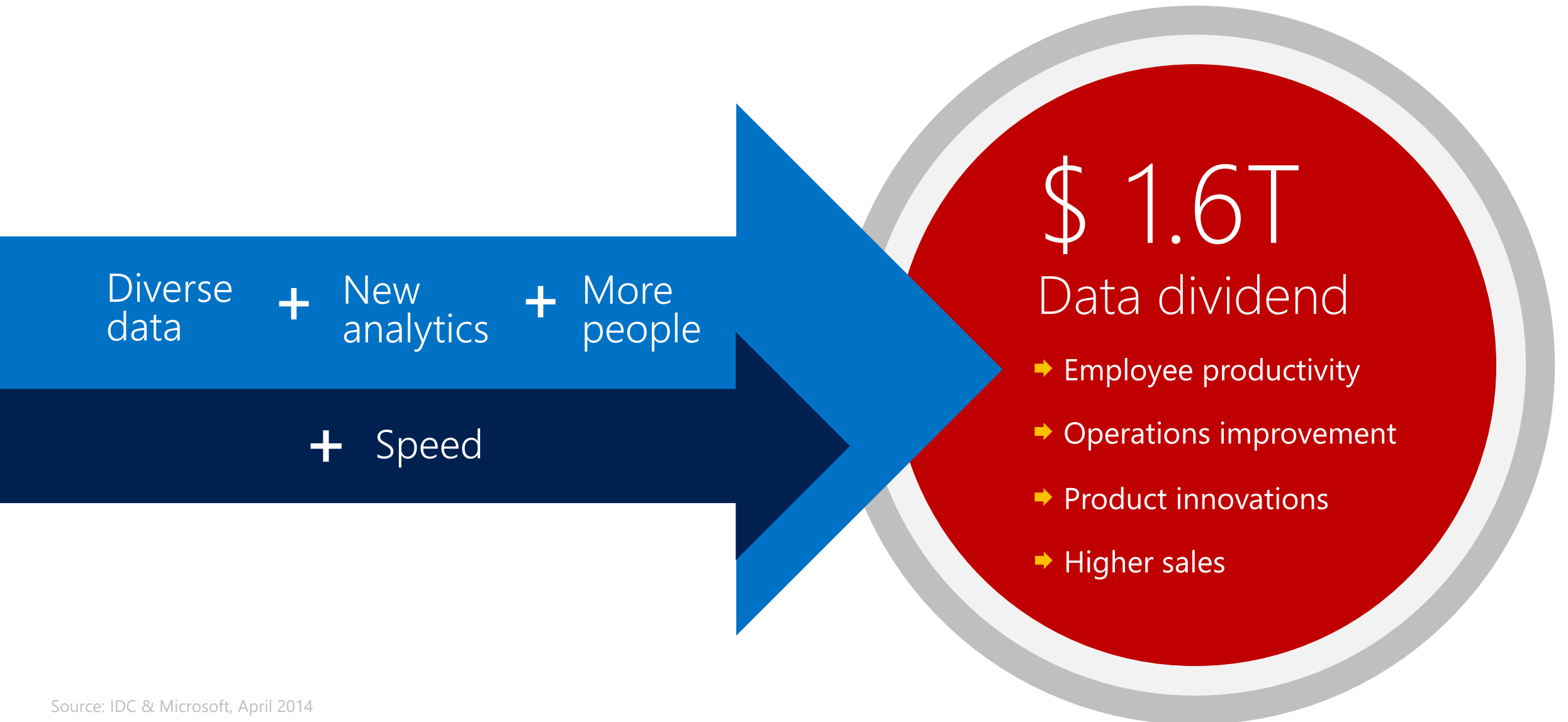
Partner Technical Consultant



Data Platform Airlift

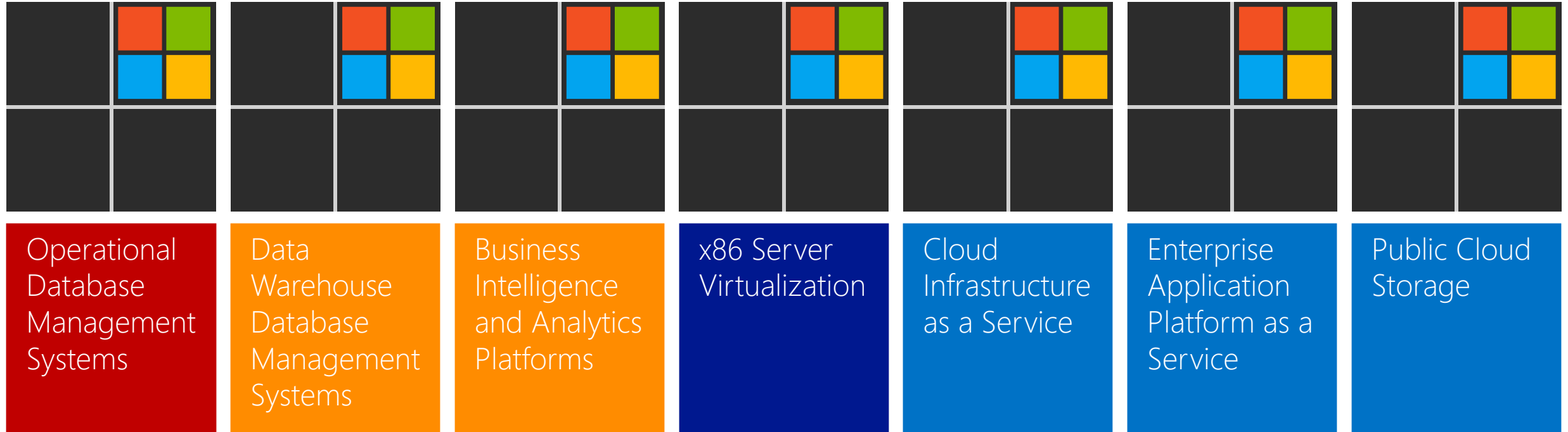
21 de Outubro \\ Microsoft Lisbon Experience

Data differentiates today's leading companies



Microsoft platform leads the way on-premises and cloud

Leader in 2014 for Gartner Magic Quadrants



Do more. Achieve more.

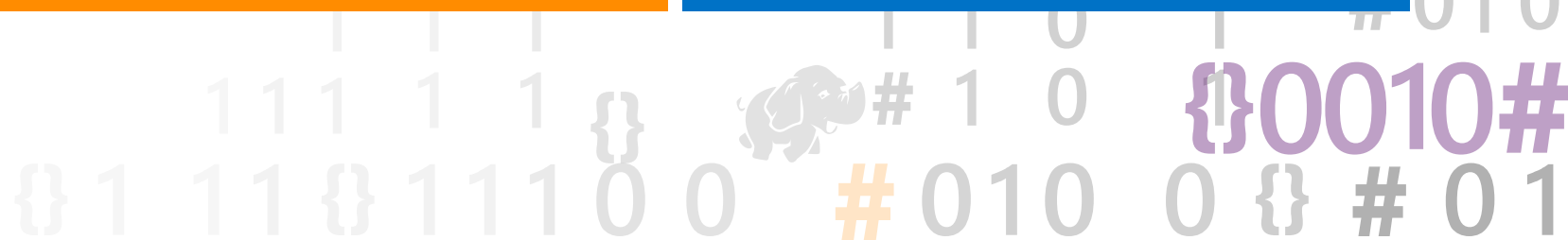
Mission critical performance



Deeper insights across data



Hyperscale cloud



Mission-critical performance

Performance

Operational analytics

Insights on operational data;
Works with in-memory OLTP and
disk-based OLTP

In-memory OLTP enhancements

Greater T-SQL surface area,
terabytes of memory supported,
and greater number of parallel
CPUs

Query data store

Monitor and optimize query plans

Native JSON

Expanded support for JSON data

Temporal database support

Query data as points in time

Security

Always encrypted

Sensitive data remains encrypted
at all times with ability to query

Row-level security

Apply fine-grained access control
to table rows

Dynamic data masking

Real-time obfuscation of data to
prevent unauthorized access

Other enhancements

Audit success/failure of database
operations

TDE support for storage of in-
memory OLTP tables

Enhanced auditing for OLTP with
ability to track history of record
changes

Availability

Enhanced AlwaysOn

Three synchronous replicas for
auto failover across domains

Round robin load balancing of
replicas

Automatic failover based on
database health

DTC for transactional integrity
across database instances with
AlwaysOn

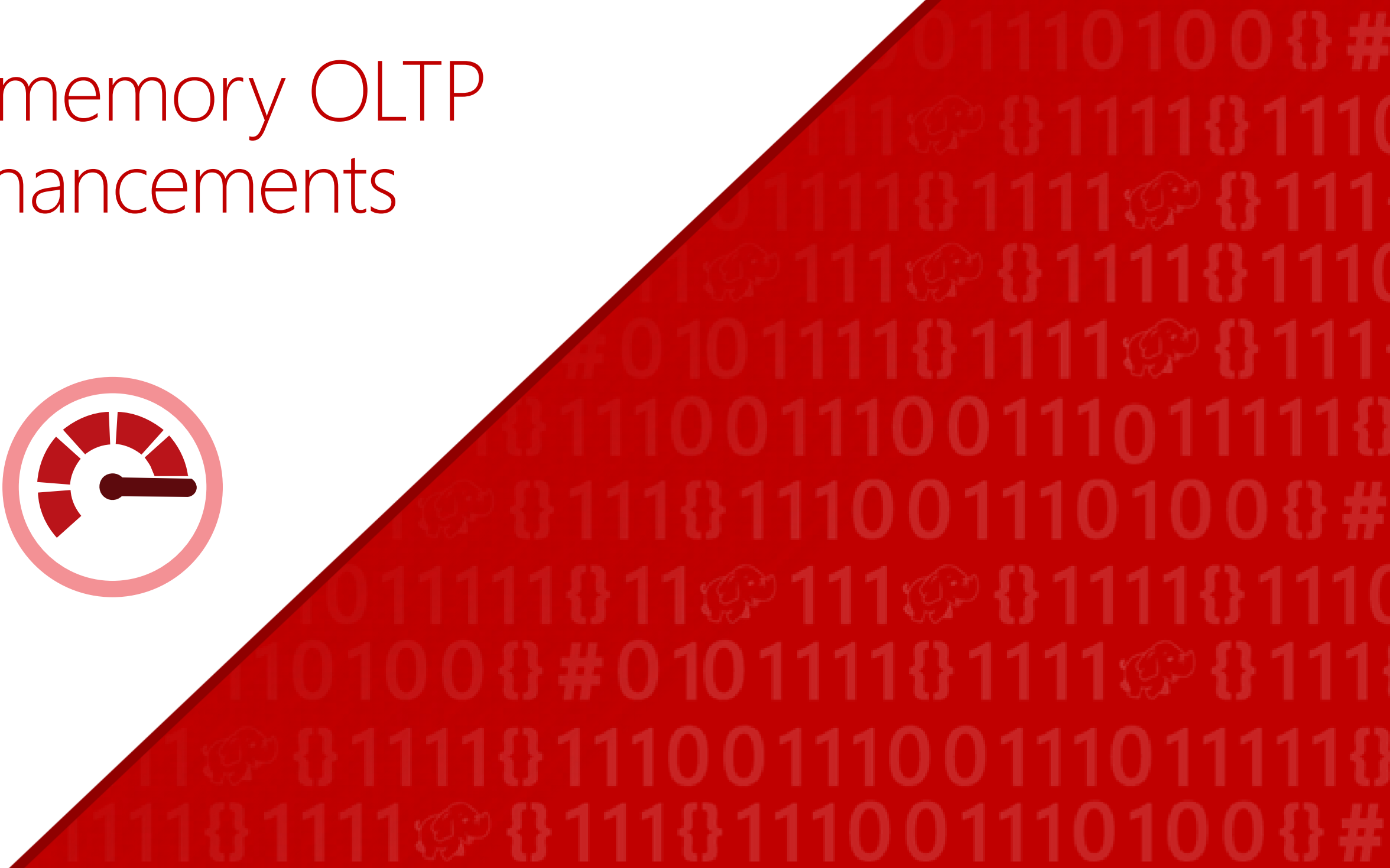
Support for SSIS with AlwaysOn

Scalability

Enhanced database caching

Cache data with automatic,
multiple TempDB files per instance
in multi-core environments

In-memory OLTP enhancements



In-memory OLTP enhancements

```
ALTER TABLE Sales.SalesOrderDetail
ALTER INDEX PK_SalesOrderID
REBUILD
WITH (BUCKET_COUNT=100000000)
```

T-SQL surface area: New

```
{LEFT|RIGHT} OUTER JOIN
Disjunction (OR, NOT)
UNION [ALL]
SELECT DISTINCT
Subqueries (EXISTS, IN, scalar)
```

ALTER support

Full schema change support: add/alter/drop column/constraint

Add/drop index supported

Surface area improvements

Almost full T-SQL coverage including scalar user-defined functions

Improved scaling

Increased size allowed for durable tables; more sockets

Other improvements

MARS support

Lightweight migration reports

Altering natively compiled stored procedures

```
CREATE PROCEDURE [dbo].[usp_1]
WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER
AS BEGIN ATOMIC WITH
(
    TRANSACTION ISOLATION LEVEL = SNAPSHOT, LANGUAGE =
N'us_english'
)
    SELECT c1, c2 from dbo.T1
END
GO

ALTER PROCEDURE [dbo].[usp_1]
WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER
AS BEGIN ATOMIC WITH
(
    TRANSACTION ISOLATION LEVEL = SNAPSHOT, LANGUAGE =
N'us_english'
)
    SELECT c1 from dbo.T1
END
GO
```

You can now perform **ALTER** operations on natively compiled stored procedures using the **ALTER PROCEDURE** statement

Use **sp_recompile** to recompile stored procedures on the next execution

Using multiple active result sets (MARS)

```
Data Source=MSSQL; Initial Catalog=AdventureWorks;  
Integrated Security=SSPI;  
MultipleActiveResultSets=True
```

Setup MARS connection for memory optimized tables using the **MultipleActiveResultSets=True** in your connection string

Query Store

Your flight data recorder
for your database



Have You Ever...?

...had your system down/slowed down and everyone waiting for you to magically fix the problem ASAP?

...upgraded an application to the latest SQL Server version and had an issue with a plan change slowing your application down?

...had a problem with your Azure SQL Database and been unable to determine what was going wrong?

With Query Store...

I **CAN** get full history of query execution

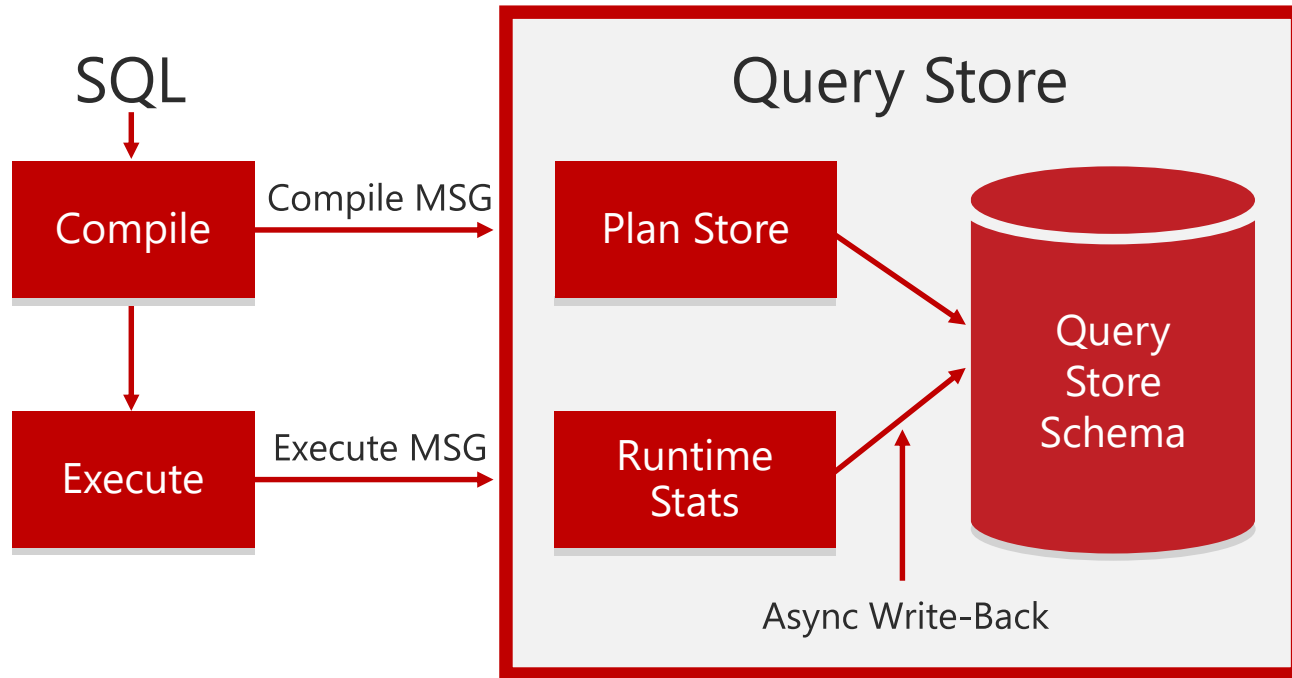
I **CAN** quickly pinpoint the most expensive queries

I **CAN** get all queries that regressed

I **CAN** easily force better plan from history with a single line of T-SQL

I **CAN** safely do server restart or upgrade

Query data store



Durability latency controlled by DB option
`DATA_FLUSH_INTERNAL_SECONDS`

Collects query texts (+ all relevant properties)

Stores all plan choices and performance metrics

Works across restarts / upgrades / recompiles

Dramatically lowers the bar for perf.
Troubleshooting

New Views

Intuitive and easy plan forcing

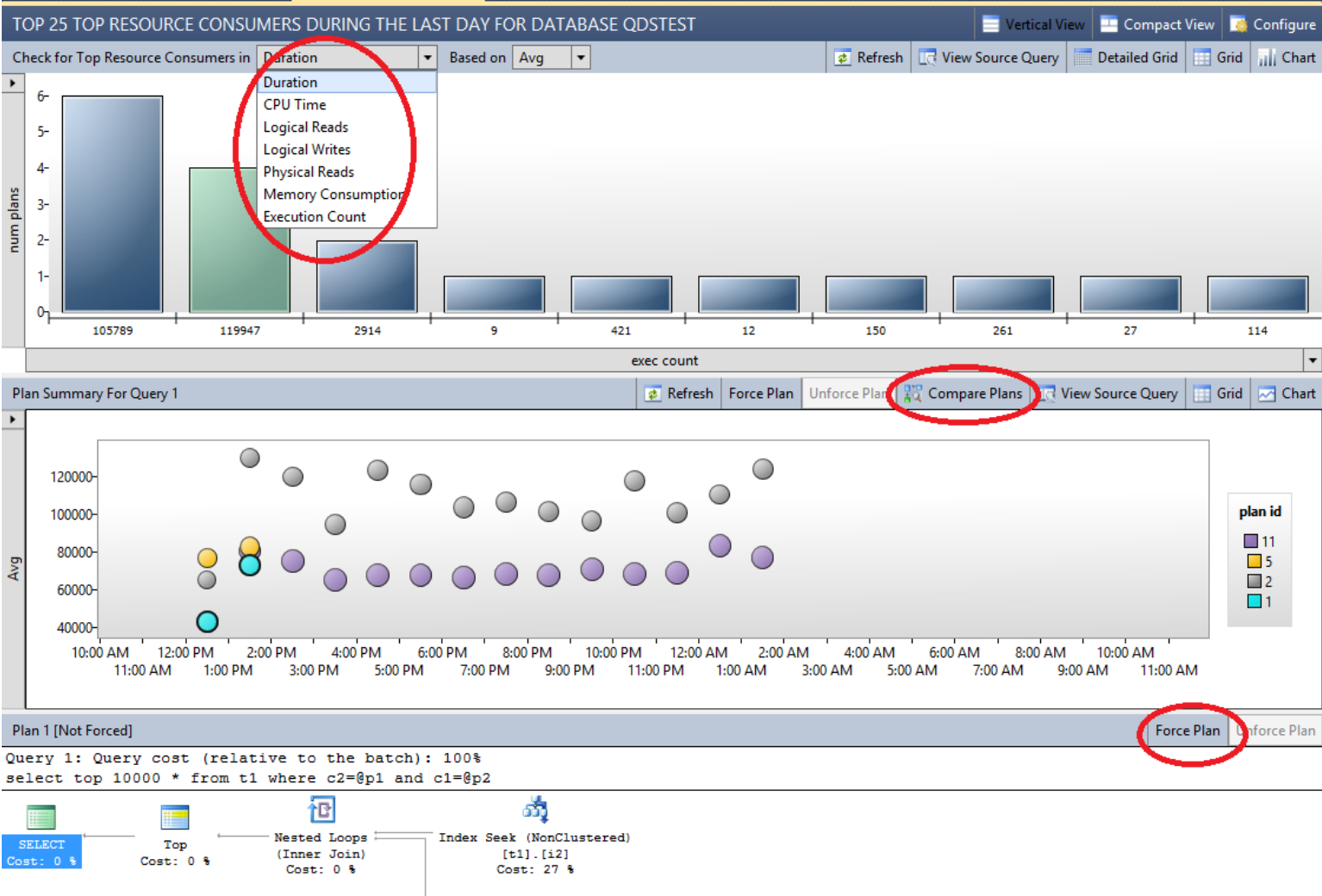
Query Store

Your flight data recorder
for your database



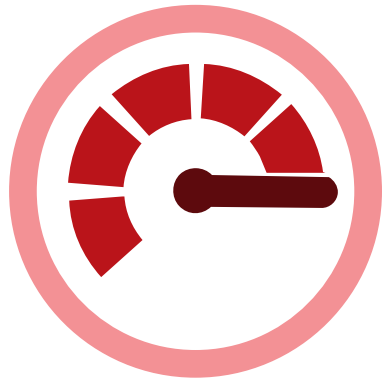
Demo

Monitoring Performance By Using the Query Store



The query store feature provides DBAs with insight on query plan choice and performance

Java Script Object Notation (JSON)



Data exchange with JSON

SELECT * FROM myTable
FOR JSON AUTO

Number	Date	Customer	Price	Quantity
SO43659	2011-05-31T00:00:00	AW29825	59.99	1
SO43661	2011-06-01T00:00:00	AW73565	24.99	3

SELECT * FROM
OPENJSON(@json)

```
[  
  {  
    "Number": "SO43659",  
    "Date": "2011-05-31T00:00:00",  
    "AccountNumber": "AW29825",  
    "Price": 59.99,  
    "Quantity": 1  
  },  
  {  
    "Number": "SO43661",  
    "Date": "2011-06-01T00:00:00",  
    "AccountNumber": "AW73565",  
    "Price": 24.99,  
    "Quantity": 3  
  }  
]
```

How to handle JSON?

No new data type

If you need to store it raw, store it as **NVARCHAR**

What is new:

Easy export: **FOR JSON**

Easy import: **OPENJSON**

Easy handling: **ISJSON, JSON_VALUE**



OPENJSON

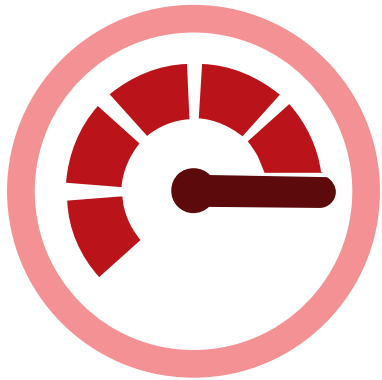
```
{ "Orders": { "OrdersArray":  
  [  
    {  
      "Order": {  
        "Number": "S043659",  
        "Date": "2011-05-31T00:00:00"  
      },  
      "AccountNumber": "AW29825",  
      "Item": {  
        "Price": 2024.9940,  
        "Quantity": 1  
      }  
    },  
    {  
      "Order": {  
        "Number": "S043661",  
        "Date": "2011-06-01T00:00:00"  
      },  
      "AccountNumber": "AW73565",  
      "Item": {  
        "Price": 2024.9940,  
        "Quantity": 3  
      }  
    }  
  ]  
} }
```

```
OPENJSON (@json, N'$.Orders.OrdersArray')  
WITH (  
  Number    varchar(200) N'$.Order.Number',  
  Date      datetime     N'$.Order.Date',  
  Customer  varchar(200) N'$.AccountNumber',  
  Quantity  int          N'$.Item.Quantity'  
)
```

Number	Date	Customer	Quantity
SO43659	2011-05-31T00:00:00	AW29825	1
SO43661	2011-06-01T00:00:00	AW73565	3

Temporal

Query back in time



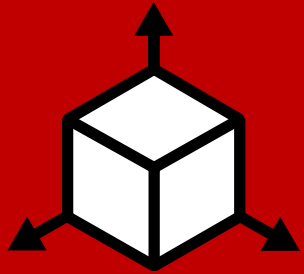
Why Temporal



Time Travel



Data Audit



Slowly Changing
Dimensions



Repair record-level
corruptions

Real data sources are dynamic

Historical data may be critical to business success

Traditional databases fail to provide required insights

Workarounds are...

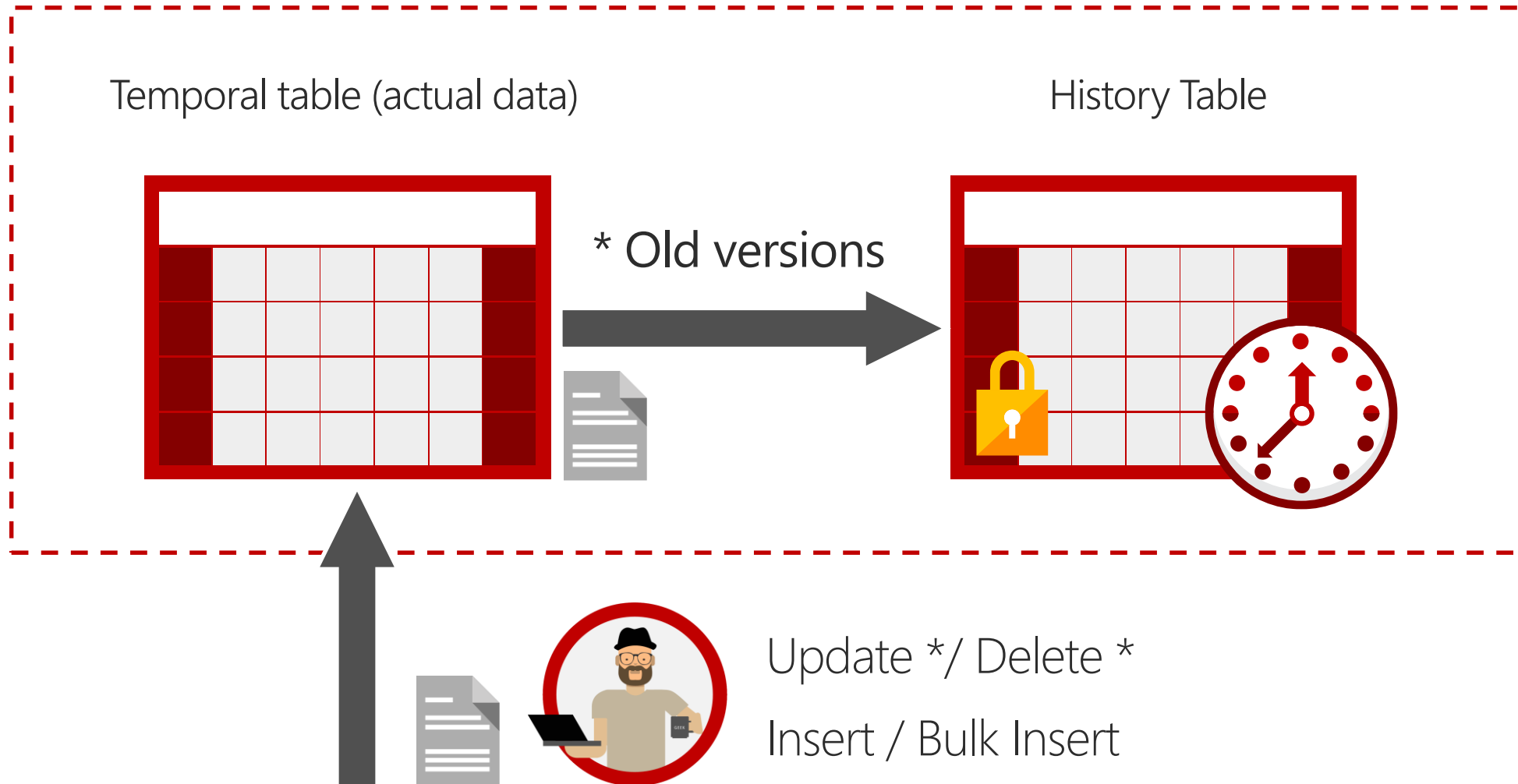
Complex, expensive, limited, inflexible, inefficient

SQL Server 2016 makes life easy

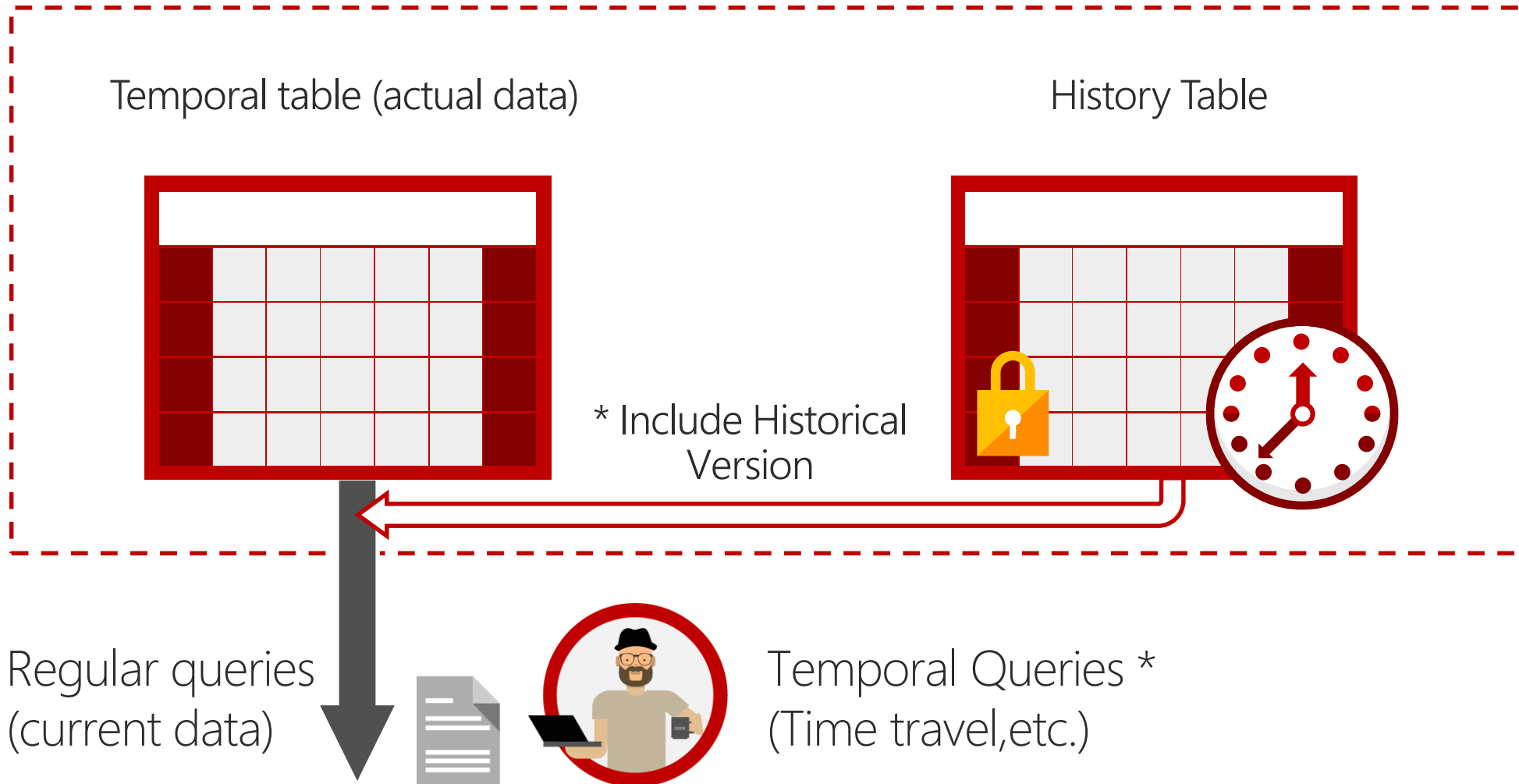
No change in programming model

New Insights

How system-time works?



How system-time works?



Dynamic Data Masking Always Encrypted Row Level Security



@14h00 / Data Platform Track

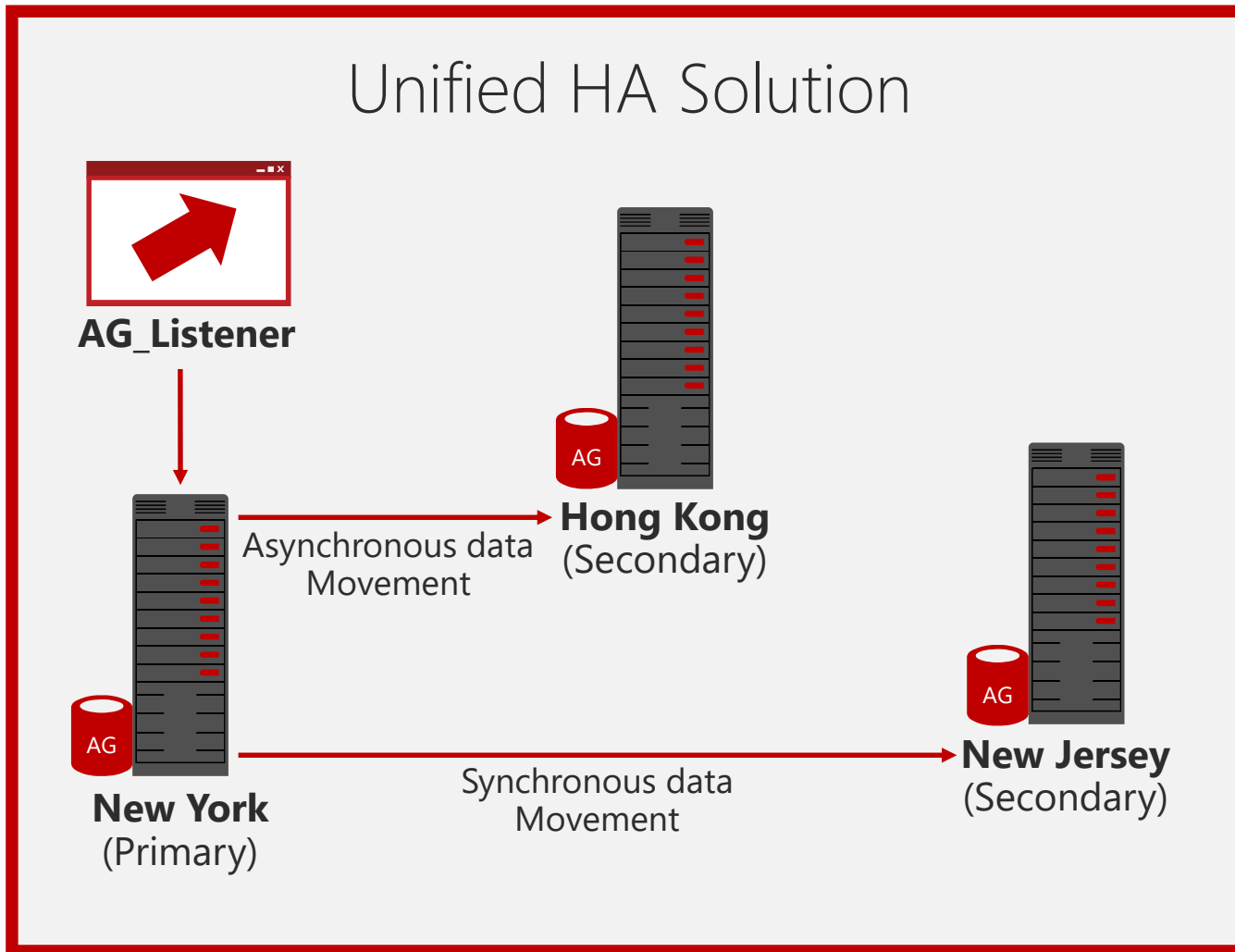
SQL Server 2016 Security - 3 wishes
were satisfied

Luís Canastreiro

Enhanced AlwaysOn



Enhanced AlwaysOn Availability Groups



Greater scalability:

Load balancing readable secondaries

Increased number of auto-failover targets

Log transport performance

Improved manageability:

DTC support

Database-level health monitoring

Group managed service account

Scalability improvements



Enhanced database caching

Supports caching data with automatic, multiple TempDB files per instance in multi-core environments

Reduces metadata and allocation contention for TempDB workloads, improving performance and scalability

By default, setup adds as many tempdb files as the CPU count or 8, whichever is lower

```
setup.exe /Q /ACTION="INSTALL" /IACCEPTSQLSERVERLICENSETERMS /FEATURES="SqlEngine"  
/INSTANCENAME="SQL15" .. /SQLTEMPDBDIR="D:\tempdb" /SQLTEMPDBFILECOUNT="4"
```

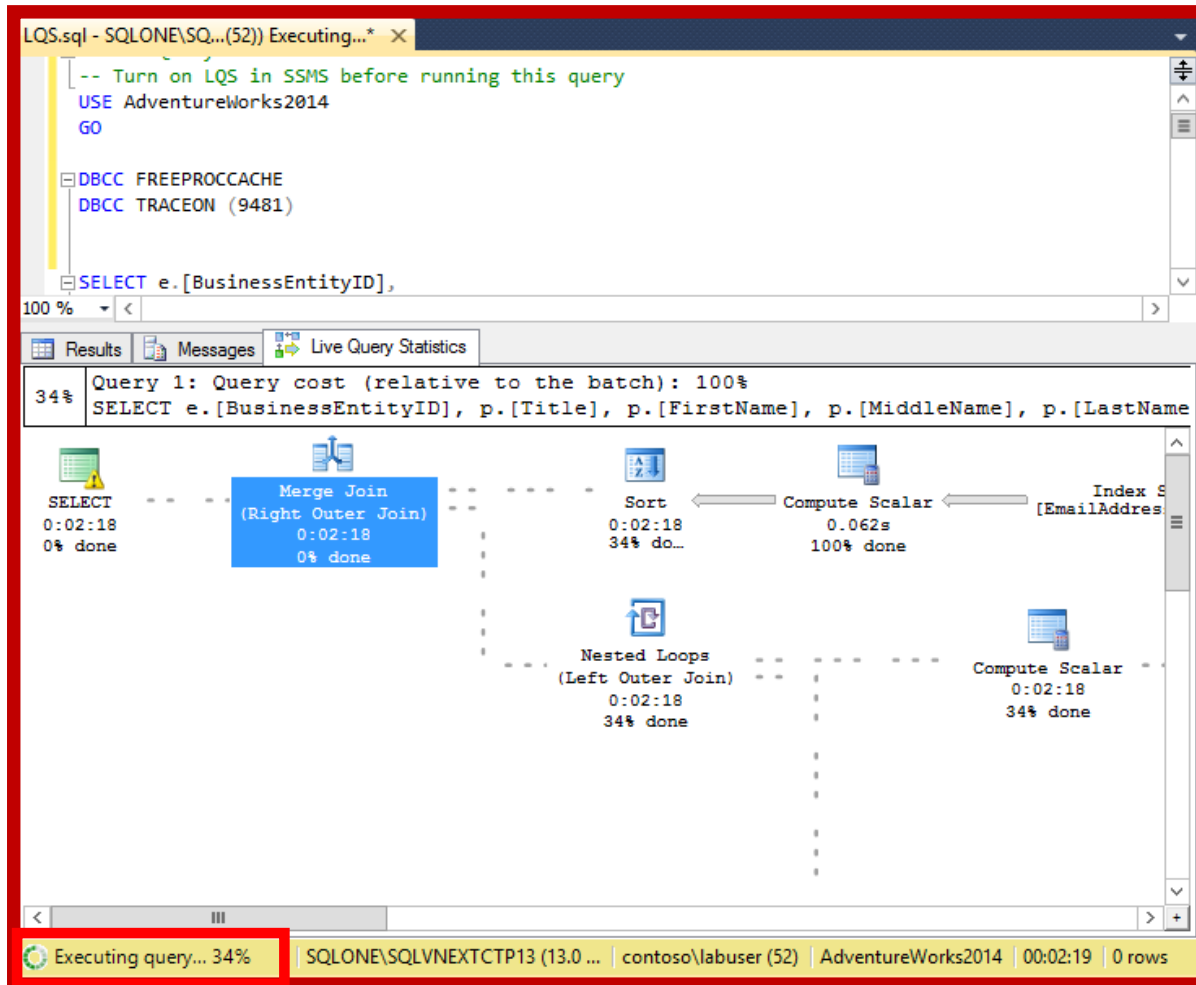
Live Query Statistics

Live query metrics



Live query statistics

Collect actual metrics about query while running



View CPU/memory usage, execution time, query progress, etc.

Enables rapid identification of potential bottlenecks for troubleshooting query performance issues.

Allows drill down to live operator level statistics:

Number of generated rows

Elapsed time

Operator progress

Live warnings, etc.

Live Query statistics



Demo

Deeper insights across data

Access any data

PolyBase

Insights from data across SQL Server and Hadoop with simplicity of T-SQL

Enhanced SSIS

Designer support for previous SSIS versions

Support for Power Query

Scale and manage

Enterprise-grade Analysis Services

Enhanced performance and scalability for analysis services

Single SSDT in Visual Studio 2015 (CTP3)

Build richer analytics solutions as part of your development projects in Visual Studio

Enhanced MDS

Excel add-in 15x faster; more granular security roles; archival options for transaction logs; and reuse entities across models

Powerful Insights

Mobile BI

Business insights for your on-premises data through rich visualization on mobile devices with native apps for Windows, iOS and Android

Enhanced Reporting Services

New modern reports with rich visualizations

Advanced analytics

R integration (CTP3)

Bringing predictive analytic capabilities to your relational database

Analytics libraries (CTP3)

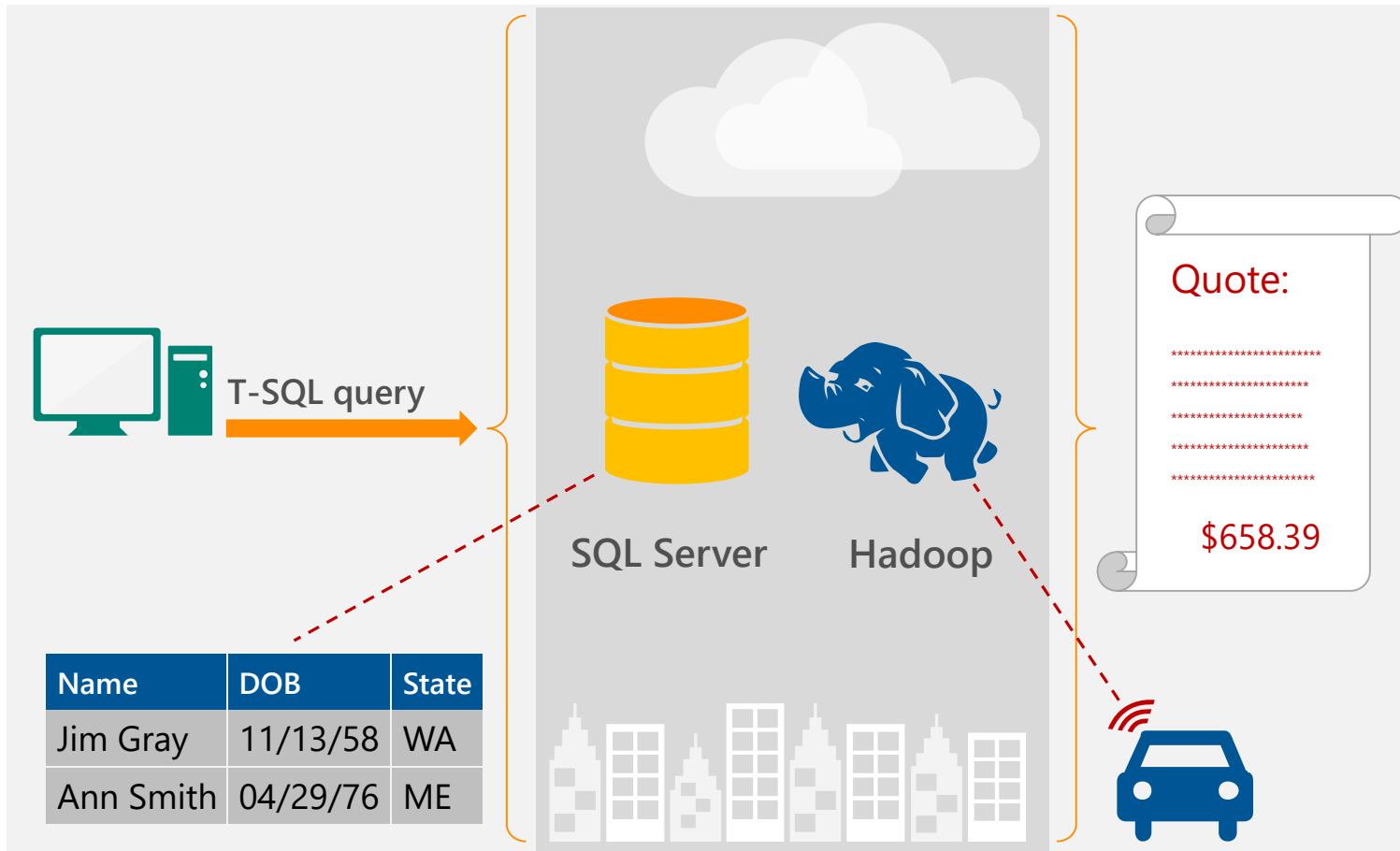
Expand your "R" script library with Microsoft Azure Marketplace

PolyBase for SQL Server 2016



PolyBase

Query relational and non-relational data with T-SQL



Capability

T-SQL for querying relational and non-relational data across SQL Server and Hadoop

Benefits

- ➔ New business insights across your data lake
- ➔ Leverage existing skillsets and BI tools
- ➔ Faster time to insights and simplified ETL process

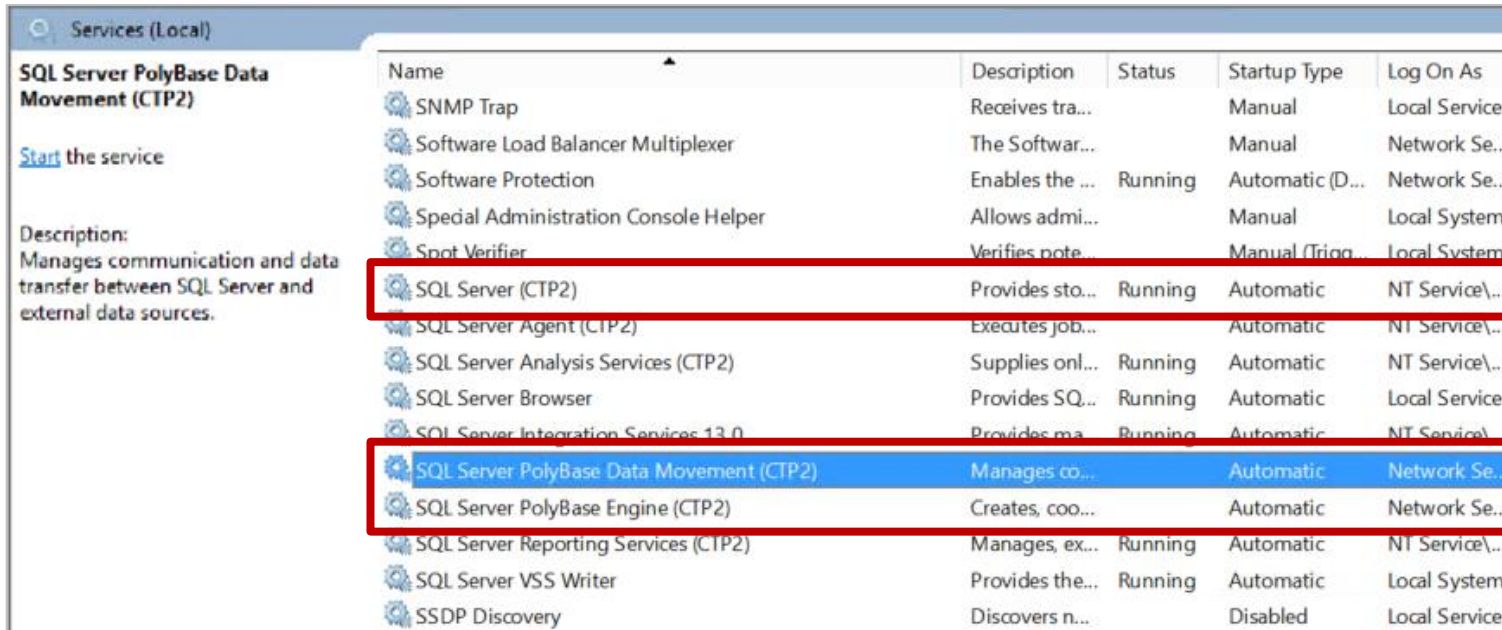
Choose Hadoop data source with sp_configure

```
-- Run sp_configure 'hadoop connectivity'  
-- and set an appropriate value  
sp_configure  
    @configname = 'hadoop connectivity',  
    @configvalue = 7;  
GO  
RECONFIGURE  
GO  
  
-- List the configuration settings for  
-- one configuration name  
sp_configure @configname='hadoop connectivity';  
GO
```

Option values

- 0: Disable Hadoop connectivity
- 1: Hortonworks HDP 1.3 on Windows Server
Azure blob storage (WASB[S])
- 2: Hortonworks HDP 1.3 on Linux
- 3: Cloudera CDH 4.3 on Linux
- 4: Hortonworks HDP 2.0 on Windows Server
Azure blob storage (WASB[S])
- 5: Hortonworks HDP 2.0 on Linux
- 6: Cloudera 5.1 on Linux
- 7: Hortonworks 2.1 and 2.2 on Linux
Hortonworks 2.2 on Windows Server
Azure blob storage (WASB[S])

Start the PolyBase services



After running for sp_configure, you must stop and restart the SQL Server engine service

Run services.msc

Find the services shown below and stop each one

Restart the services

Configure PolyBase for Azure blob storage

```
-- Using credentials on database requires enabling
-- traceflag
DBCC TRACEON(4631,-1)

-- Create a master key
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'S0me!nfo';

CREATE CREDENTIAL WASBSecret ON DATABASE WITH
    IDENTITY = 'pdw_user', Secret = 'mykey==';

-- Create an external data source (Azure Blob Storage)
-- with the credential
CREATE EXTERNAL DATA SOURCE Azure_Storage WITH
(   TYPE = HADOOP,
    LOCATION
='wasb[s]://mycontainer@test.blob.core.windows.net/pat
h',
    CREDENTIAL = WASBSecret
)
```

Type methods for providing credentials

Core-site.xml in installation path of SQL Server -
<SqlBinRoot>\Polybase\Hadoop\Conf

Credential object in SQL Server for higher security

NOTE: The syntax for a database-scoped credential (CREATE CREDENTIAL ... ON DATABASE) is temporary and will change in the next release. This new feature is documented only in the examples in the CTP2 content, and will be fully documented in the next release.

Create a reference to a Hadoop cluster

```
-- Create an external data source (Hadoop)
CREATE EXTERNAL DATA SOURCE hdp2 with (
TYPE = HADOOP,
LOCATION = 'hdfs://10.xxx.xx.xxx:xxxx',
RESOURCE_MANAGER_LOCATION='10.xxx.xx.xxx:xxxx')
```

CTP2 supports the following
Hadoop distributions

Hortonworks HDP 1.3, 2.0, 2.1, 2.2 for
both Windows and Linux

Cloudera CDH 4.3, 5.1 on Linux

Query Capabilities

Joining relational and external data

```
SELECT DISTINCT C.FirstName, C.LastName,  
C.MaritalStatus  
FROM Insurance_Customer_SQL  
INNER JOIN (
```

SQL Server table

```
SELECT * FROM SensorData_ExternalHDP WHERE  
Speed > 35  
UNION ALL  
SELECT * FROM SensorData_ExternalHDP2 WHERE  
Speed > 35  
) AS SensorD  
ON C.CustomerKey = SensorD.CustomerKey
```

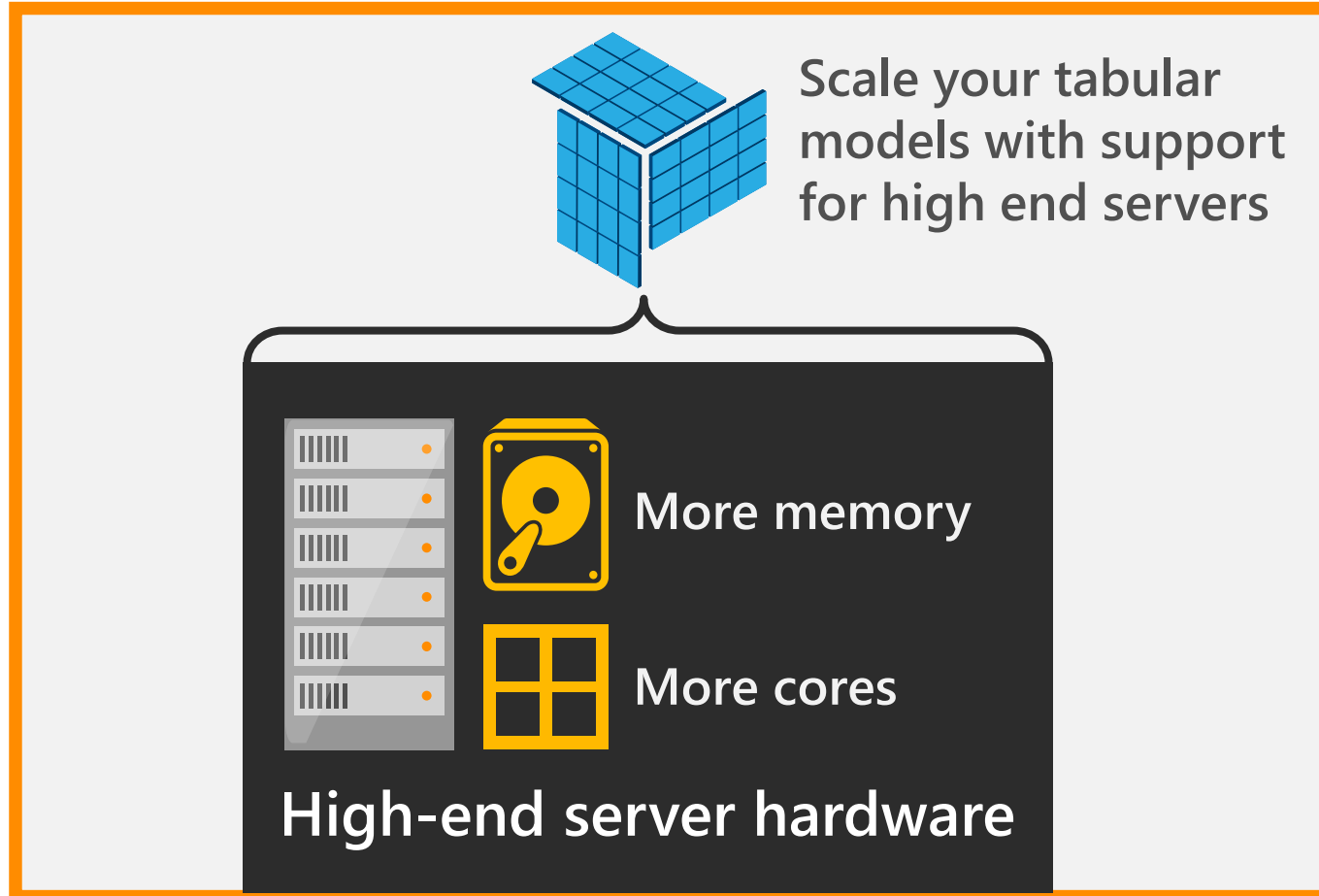
External tables
referring to data
in 2 HDP Hadoop
clusters

Enterprise grade Analysis Services



Enhanced Analysis Services

Deliver high performance and scalability for your BI solutions



Capability

Parallel partition processing

NUMA optimization for tabular models

On-demand loading and paging

Tabular and MOLAP modeling enhancements

Detect MOLAP index corruption using DBCC

Benefits

Better performance and higher scale

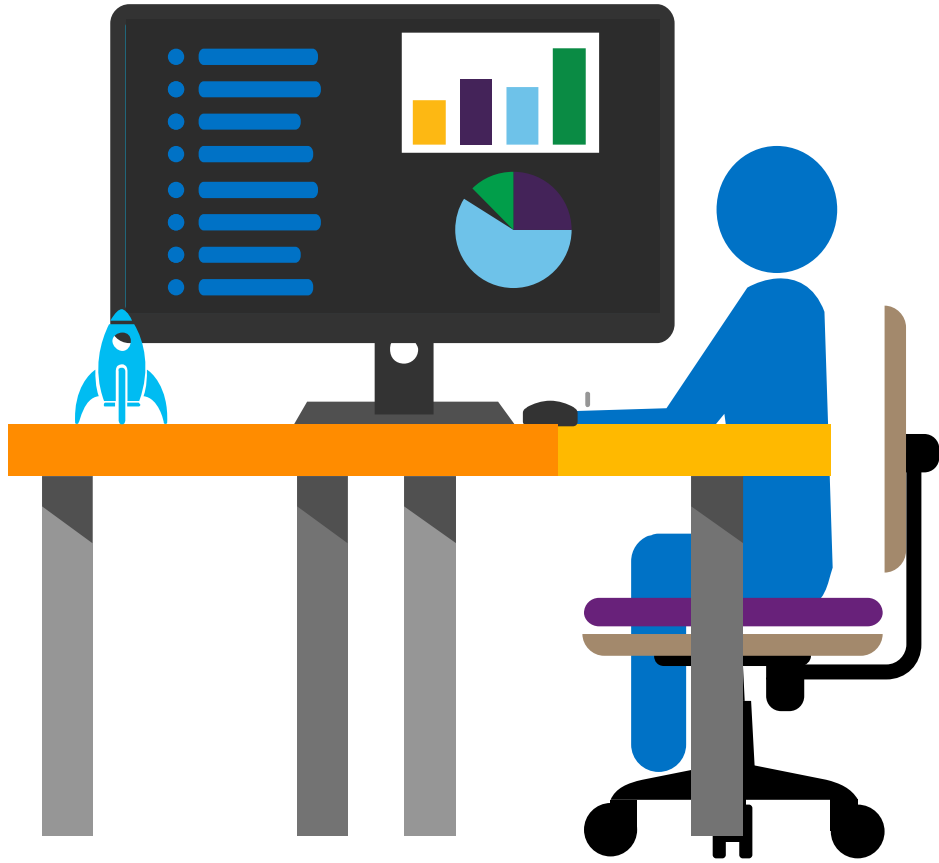
Lower TCO – more efficient utilization of existing hardware capabilities

No application changes

Enhanced Reporting Services



Modern reports with SQL Server Reporting Services



Report consumption from modern browsers

Improved parameters

Modern themes



Internet Explorer



Edge



Chrome

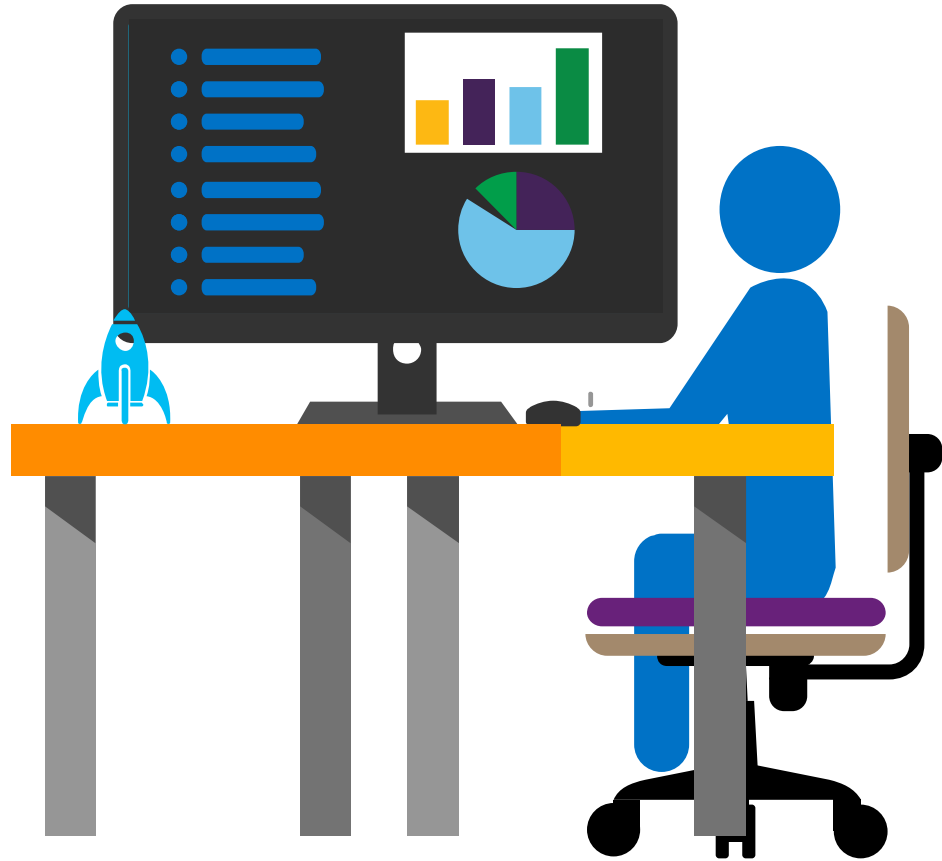


Firefox



Safari

SQL Server Reporting Services – What's new?

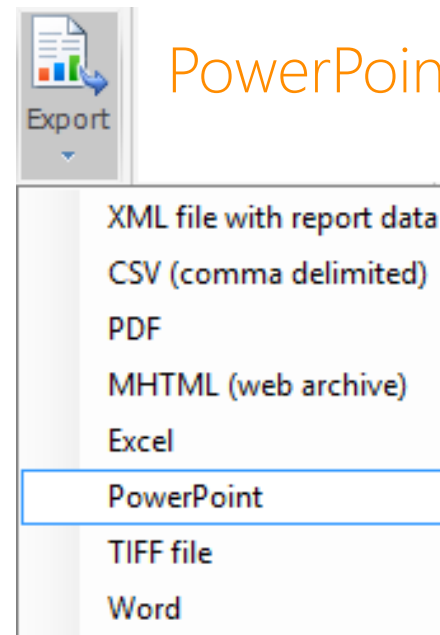


Support to .NET Framework 4

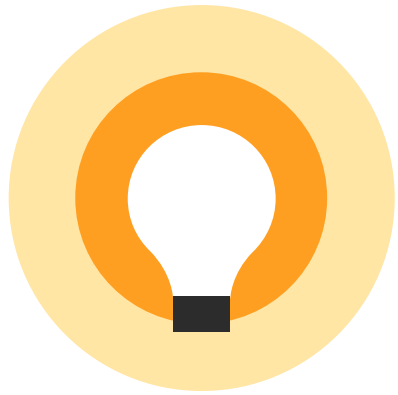
New Report Builder User Interface

HTML 5 Rendering Engine

PowerPoint Rendering and Export

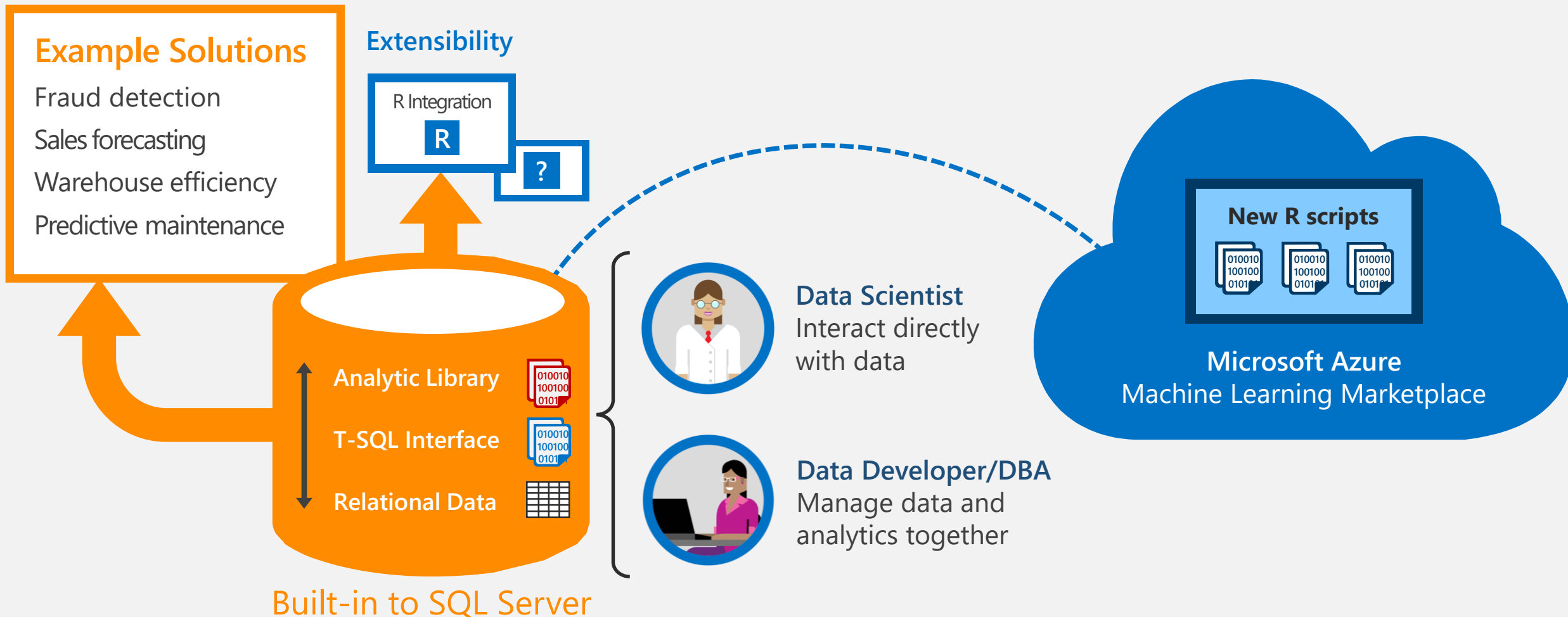


R integration with database engine



Built-in advanced analytics (CTP3)

In-database analytics



Deeper insights across data

Hybrid solutions

Stretch Database

Stretch operational tables in a secure manner into Azure for cost effective historic data availability works with Always Encrypted and Row Level Security

Power BI with on-premises data

New interactive query with Analysis Services. Customer data stays behind your firewall

Hybrid Scenarios with SSIS

Azure Data Factory integration with SSIS, package lineage and impact analysis and connect SSIS to cloud data source

Enhanced Backup to Azure

Faster restore times and 50% reduction in storage, support larger DBs with Block blobs and custom backup schedule with local staging

Simplicity

Easy migration of on-premises SQL Server

Simple point and click migration to Azure

Simplified Add Azure Replica Wizard

Automatic listener configuration for AlwaysOn in Azure VMs

Consistency

Common development, management and identity tools

Including Active Directory, Visual Studio, Hyper-V and System Center

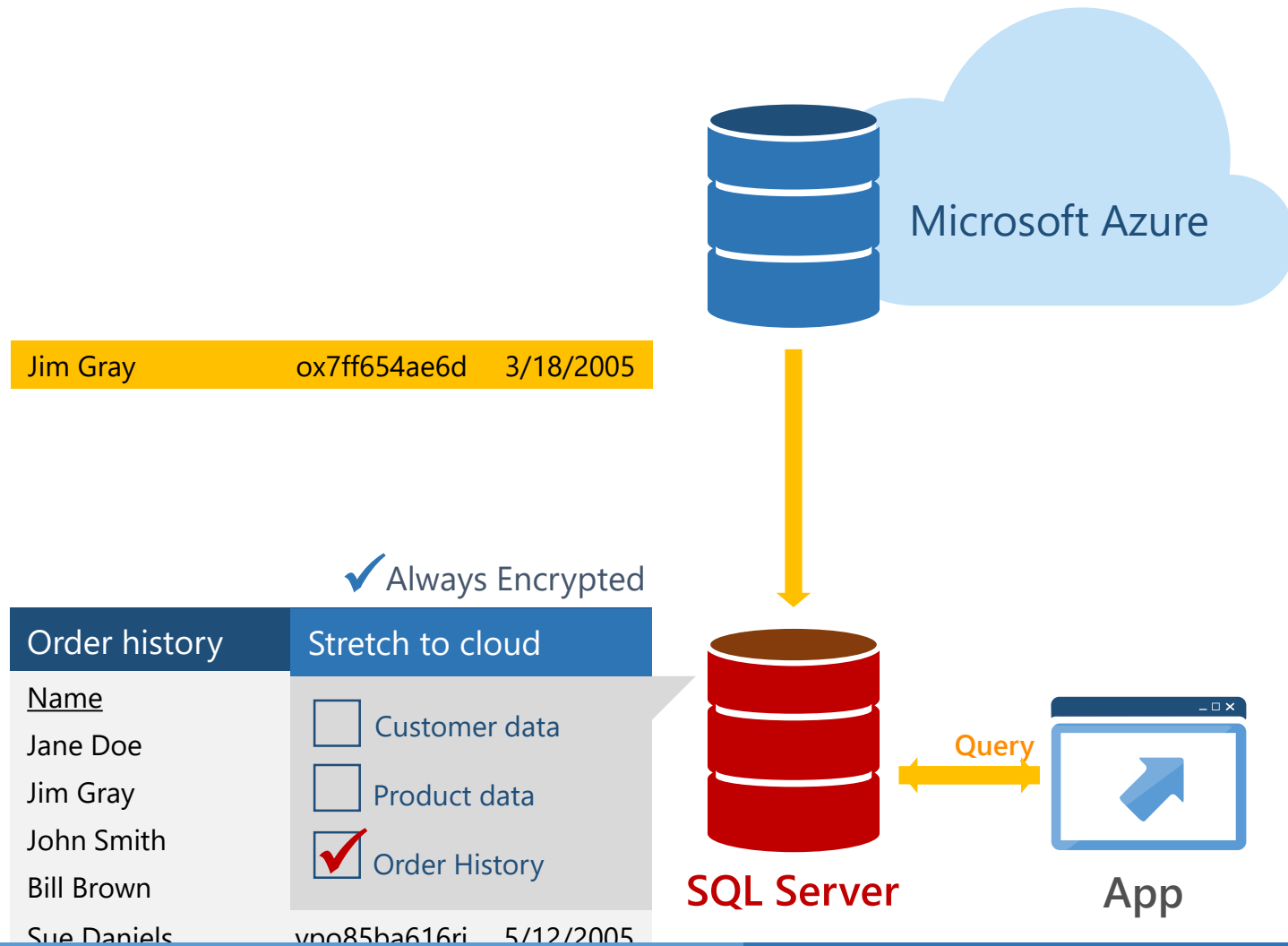
Consistent Experience from SQL Server on-premises to Microsoft Azure IaaS and PaaS

Stretch Database



Stretch SQL Server into Azure

Stretch warm and cold tables to Azure with remote query processing



Capability

Stretch cold database tables from on-premises SQL Server Databases to Azure with remote query processing

Benefits

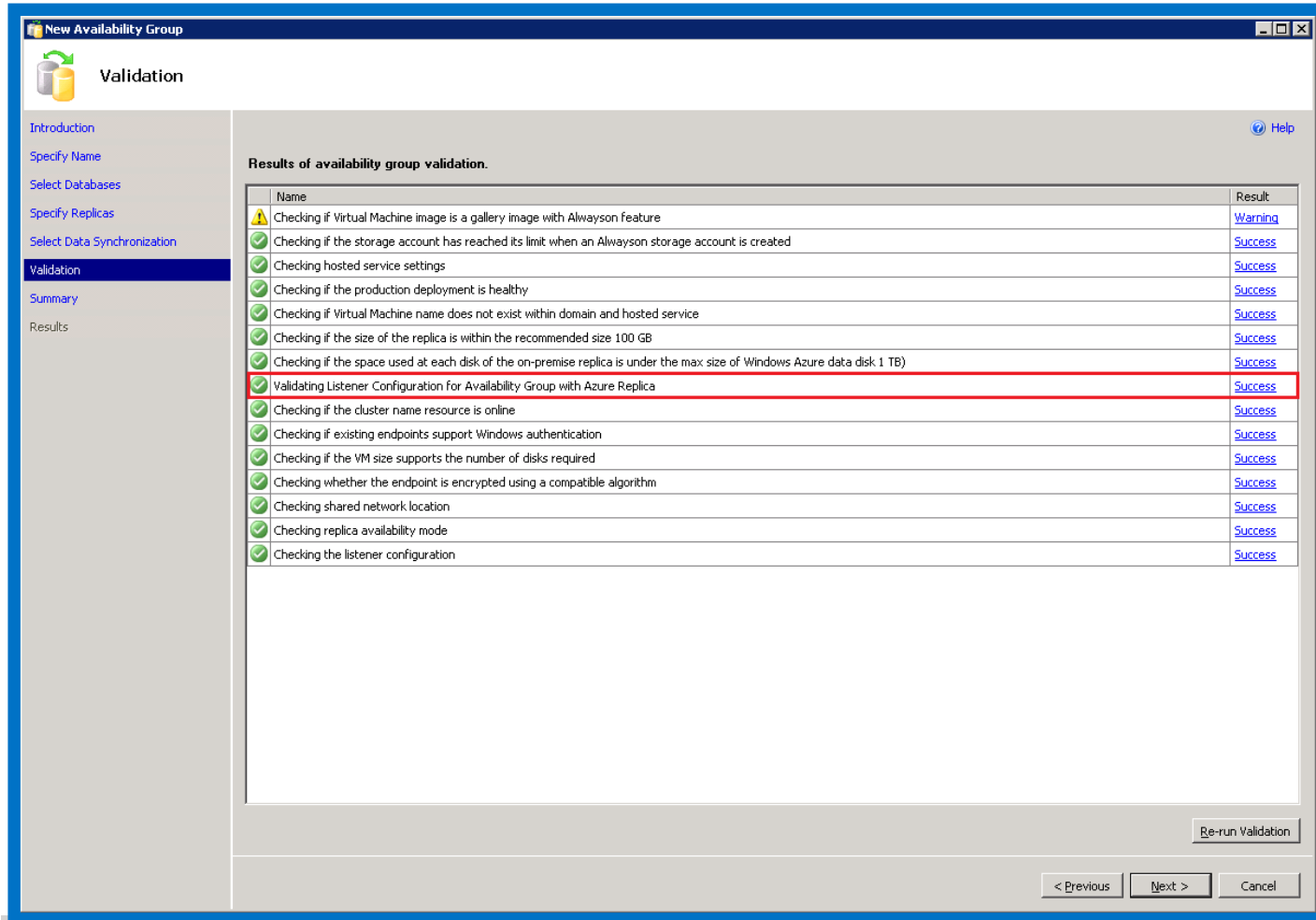
- ➔ Cost effective historical data
- ➔ Entire table is online and remains queryable from on-premises apps
- ➔ Transparent to applications
- ➔ Supports Always Encrypted & Row Level Security

Simplified
AlwaysOn with
replicas on Azure



Simplified Add Azure Replica Wizard

Automatic Listener Configuration



Today this requires manually configuring the Listener

SQL Server 2016

Allows configuring the Availability Group Listener in Azure

Clients can connect to the Azure replica after failover using the Listener name

The Microsoft data platform

 Windows Azure

 Office

Microsoft
SQL Server

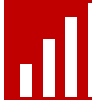
VISUALIZE + DECIDE



Apps



Reports



Dashboards



Ask



Mobile

TRANSFORM + ANALYZE



Orchestration



Extract, transform,
load



Information
management



Prediction

COLLECT + MANAGE



Relational



Non-relational



Analytical



Streaming



Internal &
external



César Mendes

cmendes@microsoft.com





Free Azure Trial

<http://aka.ms/tryazure>



Try SQL Server 2016 CTP2

<http://aka.ms/trysql2016>



Use Power BI for Free

<http://powerbi.microsoft.com>





© 2015 Microsoft Corporation. All rights reserved. Microsoft, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.