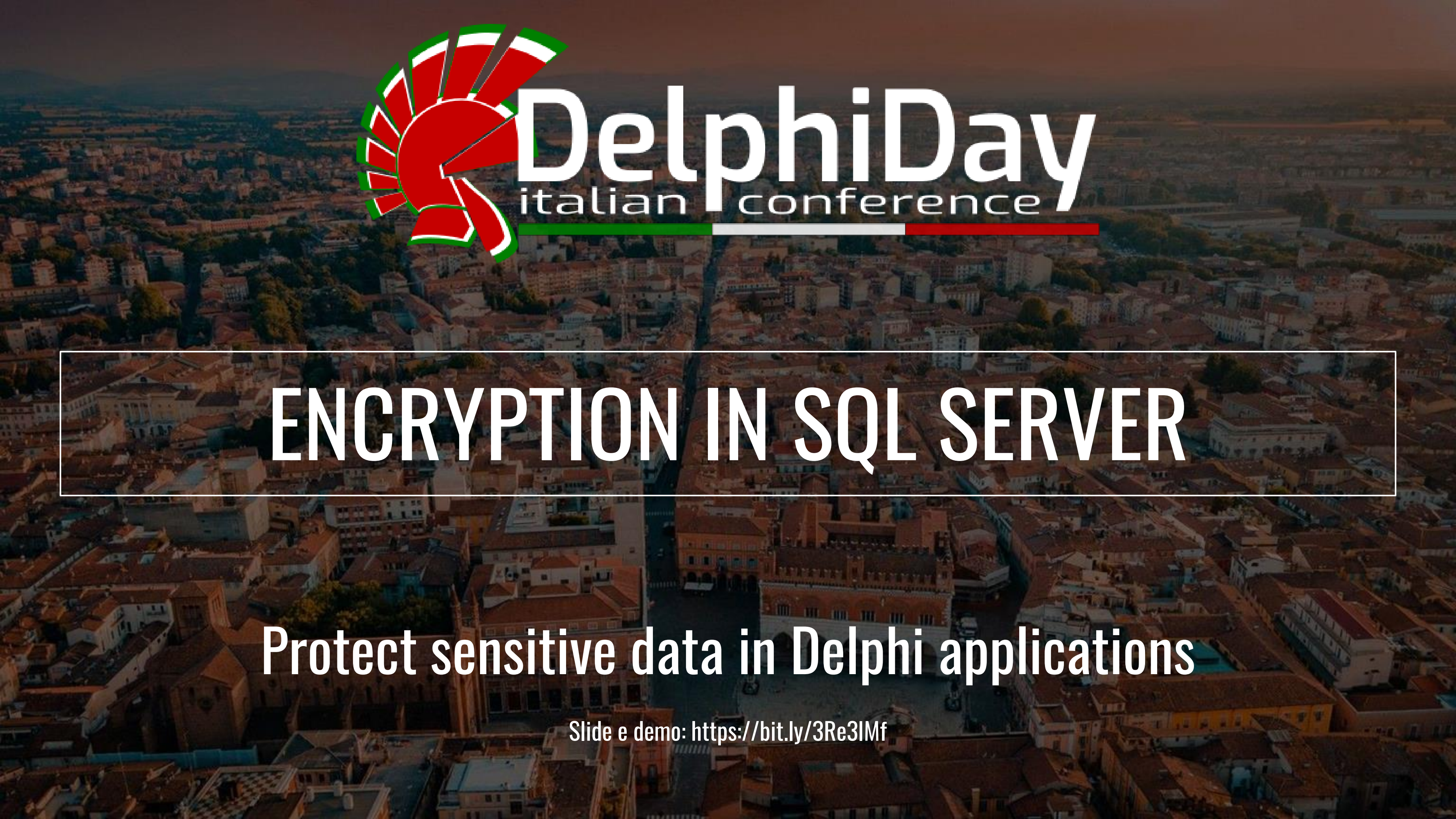


DelphiDay italian conference

ENCRYPTION IN SQL SERVER

Protect sensitive data in Delphi applications

Slide e demo: https://bit.ly/3Re3IMf

SERGIO GOVONI

CENTRO SOFTWARE S.P.A.

segovoni.medium.com

bit.ly/sgovoni-MVP

twitter.com/segovoni

github.com/segovoni

linkedin.com/in/sgovoni

11-12 Giugno 2024
Piacenza

OPEN-SOURCE PROJECTS

github.com/segovoni

SQL command-line utility

github.com/segovoni/sqlcmdcli

Alter column with dependencies

github.com/segovoni/sp_alter_column

Delphi Secure SQL Database

github.com/segovoni/DelphiSecureSQLDatabase

AGENDA

- ➔ Starting point
- ➔ Encryption in SQL Server (overview)
- ➔ SQL Server Always Encrypted
- ➔ Manage Always Encrypted columns in Delphi

Starting point

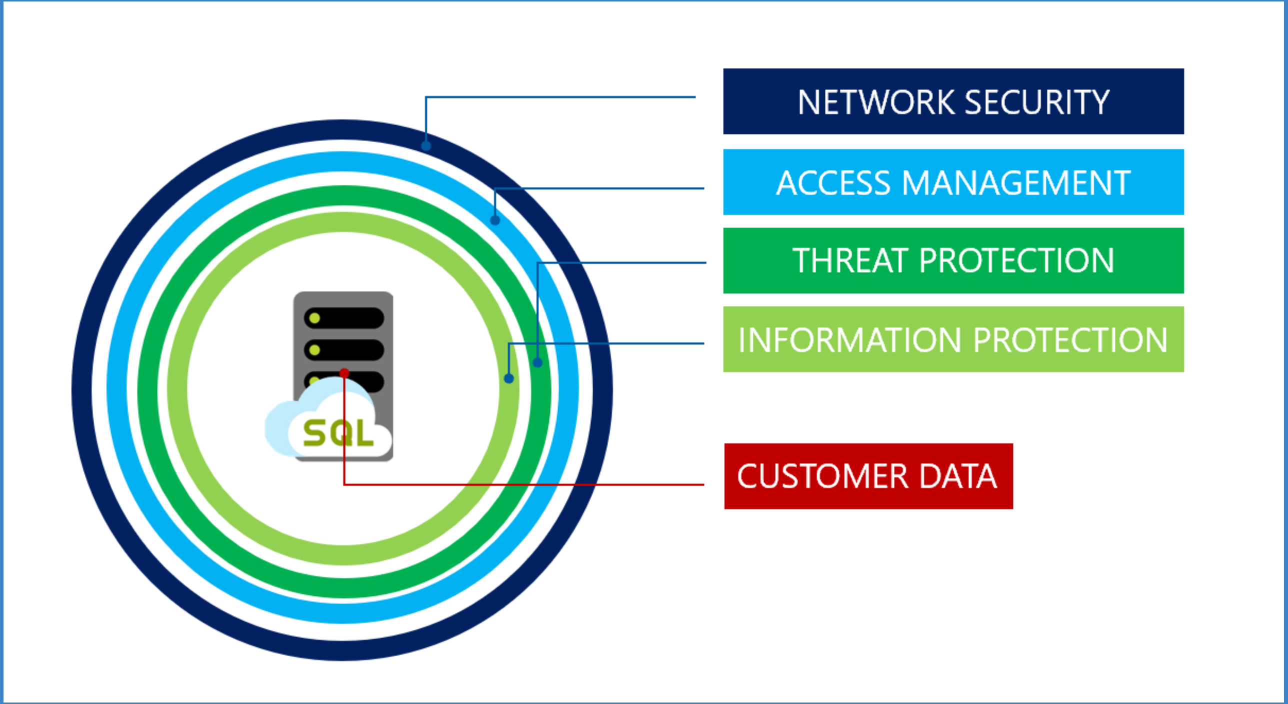
1

DelphiDay italian conference

Starting point

- ➔ GDPR
  - ➔ May 24, 2016
  - ➔ Applies from May 25, 2018
  - ➔ It doesn't come from a technical issue
  - ➔ Assessment and gap analysis
  - ➔ Mapping functional remediations with technical aspects
  - ➔ You don't have to protect/encrypt everything
- ➔ Trade secret

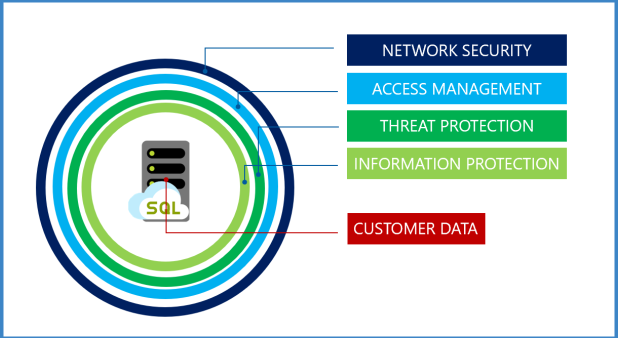
SQL Server security layers



https://learn.microsoft.com/azure/azure-sql/database/security-overview?view=azuresql

SQL Server security layers

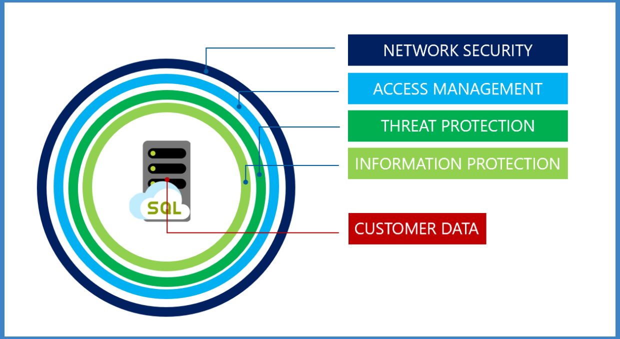
- ➔ Network security
  - ➔ Physical firewall
  - ➔ Service firewall/Virtual Network Firewall



SQL Server security layers

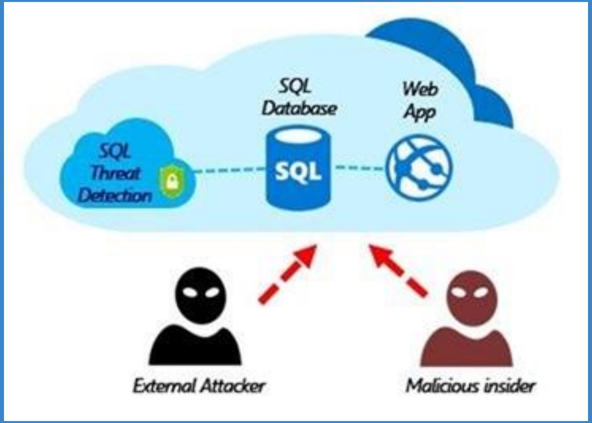
- ➔ Access Management
  - ➔ Firewall, Azure Database Firewall (device)
  - ➔ Encrypted Authentication (who am I)
  - ➔ Authorization (what can I do)

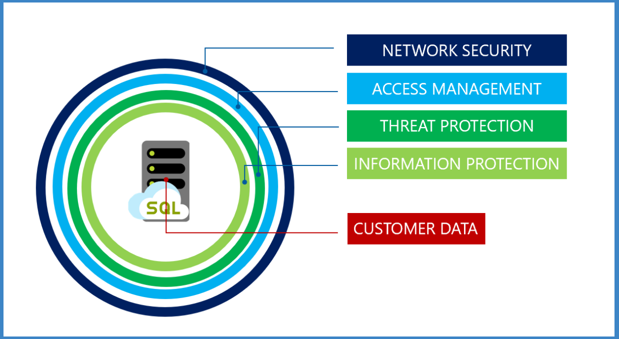
  - ➔ Less privilege principle



SQL Server security layers

- ➔ Threat Protection
  - ➔ SQL Auditing
  - ➔ Advanced Threat Protection



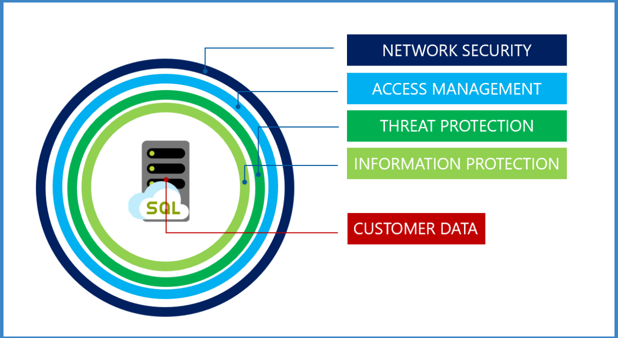


DelphiDay italian conference

SQL Server security layers

- → Information Protection
  - → Transport Layer Security (in transit)
  - → Transparent Data Encryption (at rest)

  - → Cell-Level Encryption (at rest)
  - → Always Encrypted (at rest and in transit)
  - → Dynamic Data Masking
  - → Row-Level Security



DelphiDay italian conference

Authentication

- There are two possible modes
  - Windows Authentication mode (server login)
  - Mixed mode (server login and database user)
- Windows Authentication
  - Always available, it can't be disabled
  - Kerberos
- SQL Server Authentication
  - Encrypted in all SQL Server versions
  - Self signed certificate or company certificate

Authorization

- Every SQL Server securable has associated permissions that can be granted to a principal
  - GRANT, REVOKE, and DENY
  - Hierarchical inheritance
  - Row-Level Security
- Server and database level permissions/roles
- Less privilege principle
  - EXECUTE AS

Encryption in SQL Server

2

DelphiDay
italian conference

Communication encryption

- ➔ SQL Server can use TLS to encrypt data that is transmitted across a network
- ➔ On-premise
  - ➔ Self-signed certificate
  - ➔ Certificate issued by an internal CA
  - ➔ Certificate issued by a commercial CA
  - ➔ Server and client installation for CA certificates

Communication encryption

- ➔ Azure SQL always enforces encryption (SSL/TLS) for all connections
  - ➔ All data is encrypted “in transit” between the client and server irrespective of the setting of Encrypt or TrustServerCertificate in the connection string
  - ➔ TLS 1.2
- ➔ SQL Server 2022 introduces support for TLS 1.3

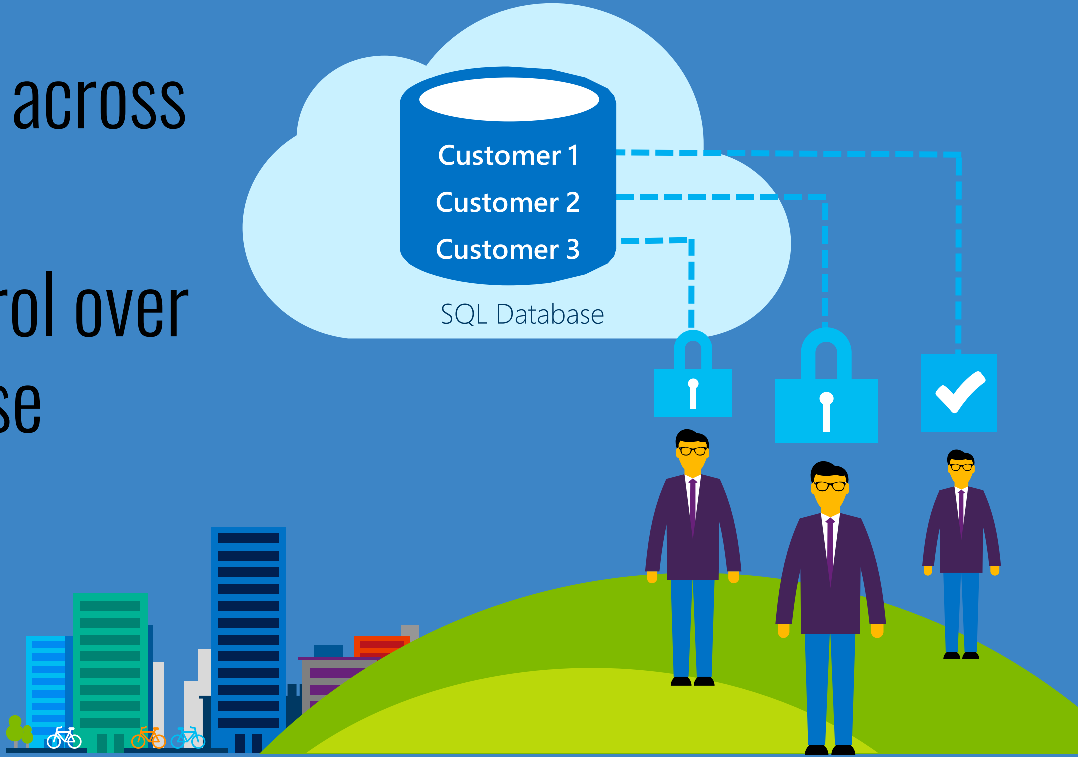
Better security from SQL 2016

- ➔ SQL Server 2016 introduces three new security features
  - ➔ Row-Level Security
  - ➔ Dynamic Data Masking
  - ➔ Always Encrypted

Row-Level Security

Row-Level Security

- ➔ Protect data privacy by ensuring the right access across rows
- ➔ Fine-grained access control over specific rows in a database table

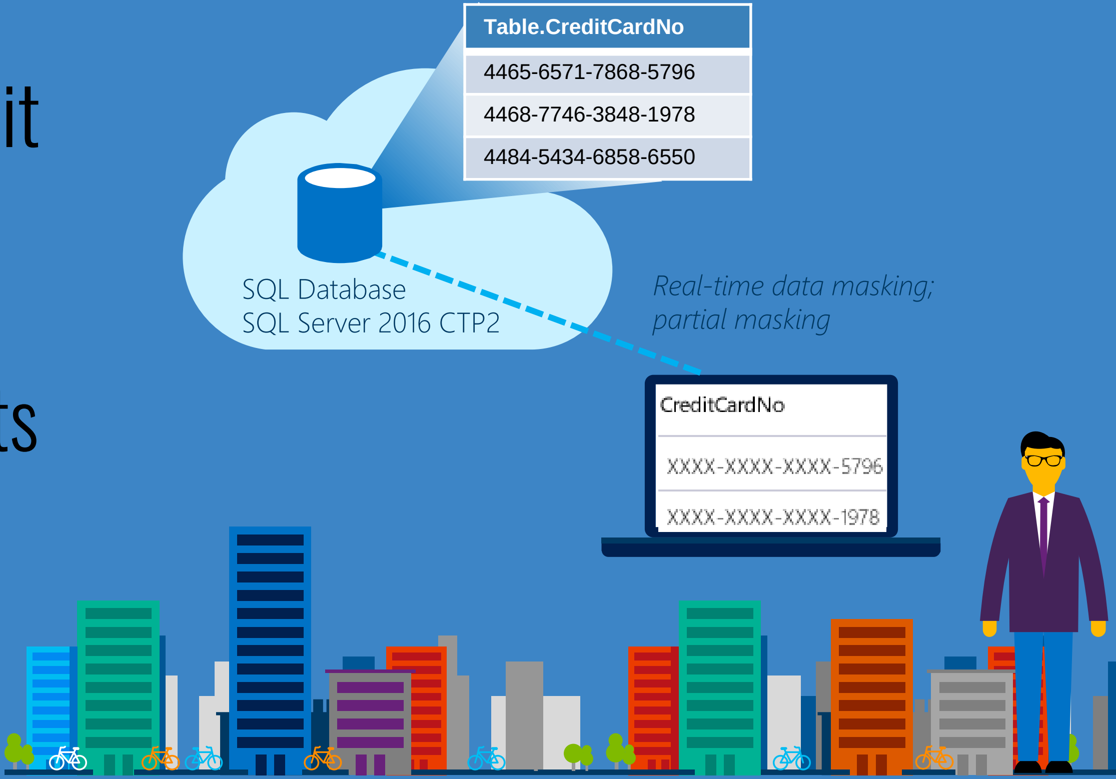


DelphiDay italian conference

Dynamic Data Masking

Dynamic Data Masking

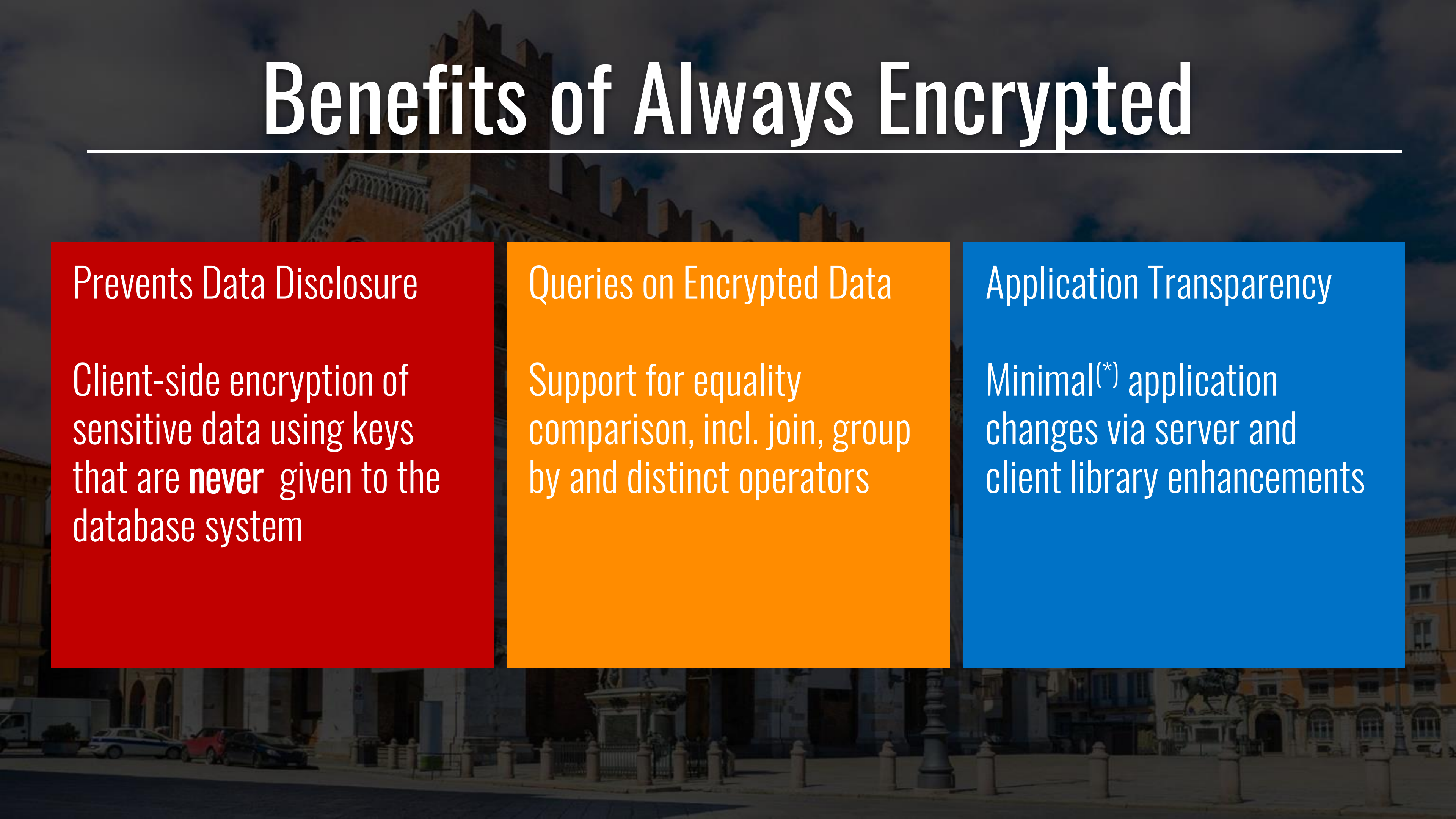
- ➔ Prevent the abuse of sensitive data by hiding it from users
- ➔ Data masking applied in real-time to query results based on policy



DelphiDay italian conference

Always Encrypted

Benefits of Always Encrypted



Prevents Data Disclosure

Client-side encryption of sensitive data using keys that are **never** given to the database system

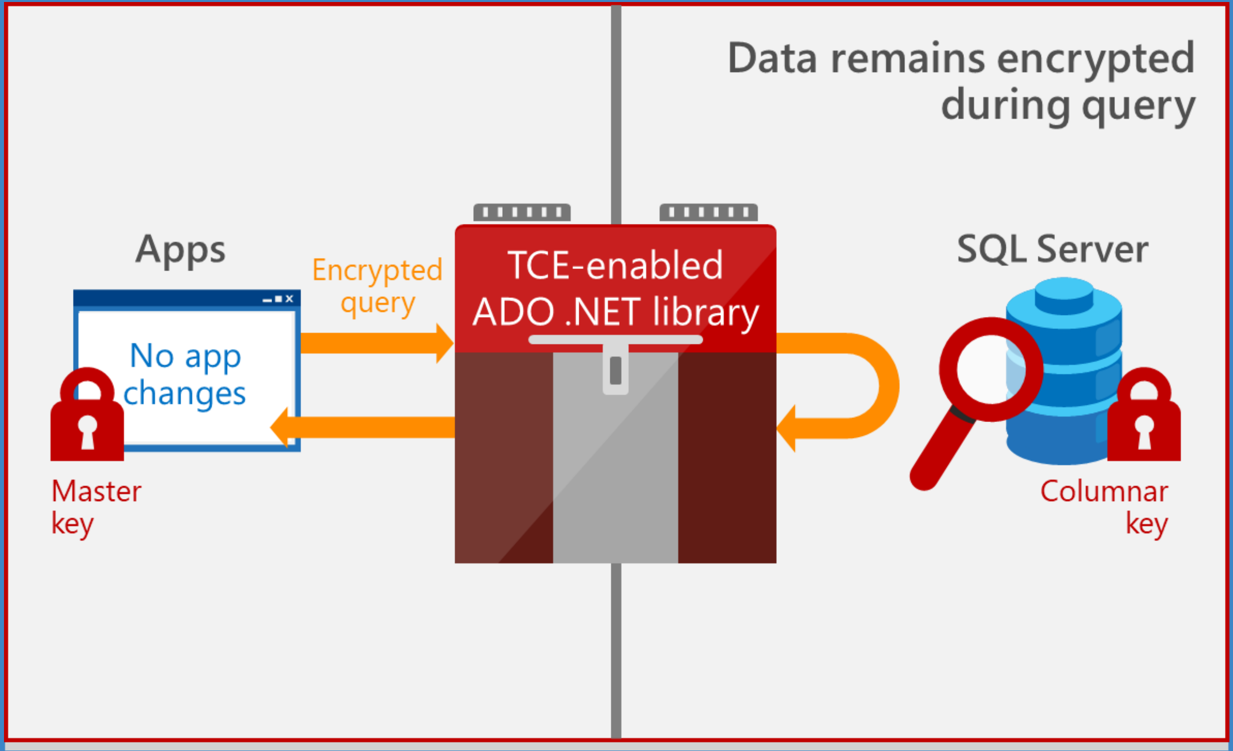
Queries on Encrypted Data

Support for equality comparison, incl. join, group by and distinct operators

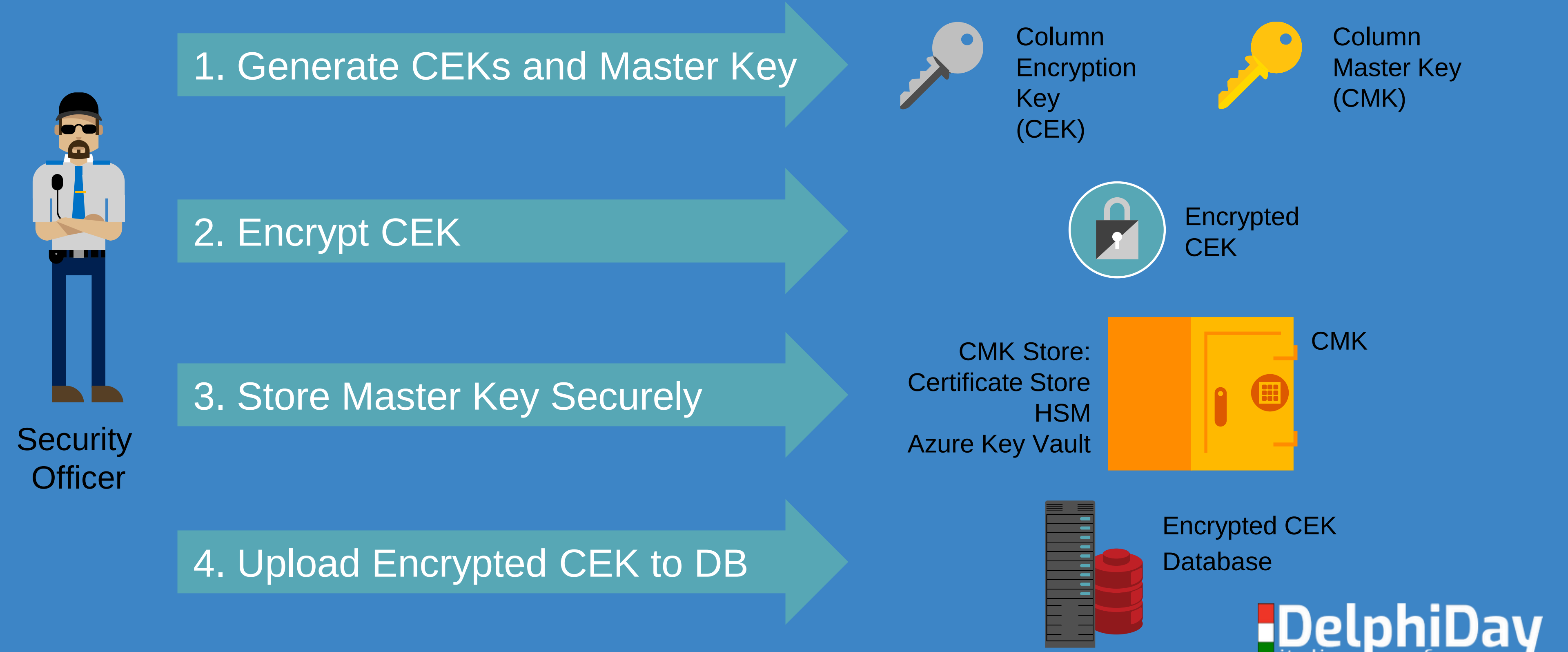
Application Transparency

Minimal(*) application changes via server and client library enhancements

Always Encrypted overview

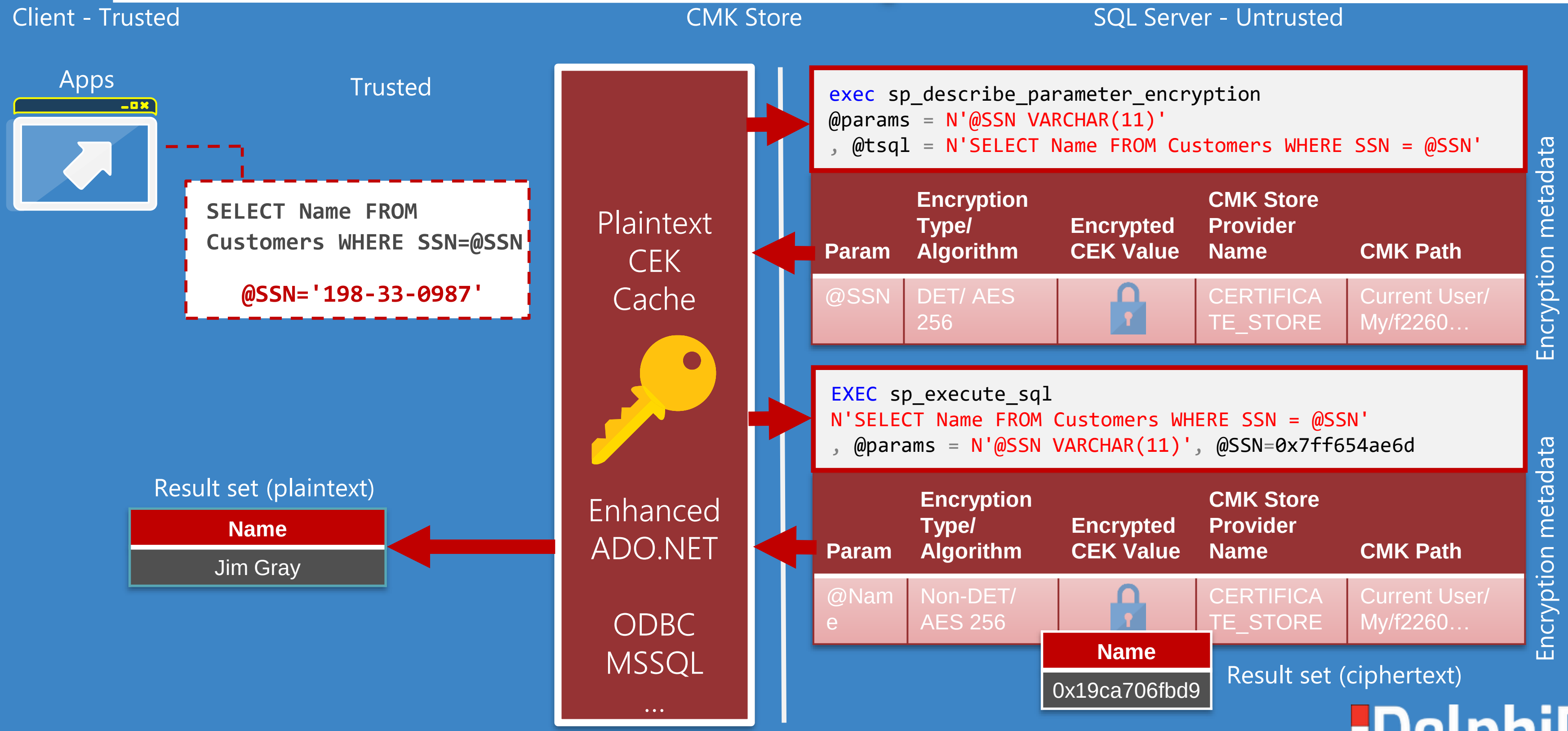


Always Encrypted Key Provisioning



DelphiDay italian conference

Example



```
SELECT Name FROM
Customers WHERE SSN=@SSN

@SSN='198-33-0987'
```

```
exec sp_describe_parameter_encryption
@params = N'@SSN VARCHAR(11)'
, @tsql = N'SELECT Name FROM Customers WHERE SSN = @SSN'
```

Param	Encryption Type/ Algorithm	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@SSN	DET/ AES 256	🔒	CERTIFICATE_STORE	Current User/ My/f2260…

```
EXEC sp_execute_sql
N'SELECT Name FROM Customers WHERE SSN = @SSN'
, @params = N'@SSN VARCHAR(11)', @SSN=0x7ff654ae6d
```

Param	Encryption Type/ Algorithm	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@Name	Non-DET/ AES 256	🔒	CERTIFICATE_STORE	Current User/ My/f2260…

Result set (ciphertext)

Name
0x19ca706fbd9

Result set (plaintext)

Name
Jim Gray

DelphiDay italian conference

Encryption types for Always Encrypted

```
Randomized encryption
   Encrypt('123-45-6789') = 0x17cfd50a
   Repeat: Encrypt('123-45-6789') = 0x9b1fcf32
   Allows for transparent retrieval of encrypted
   data but NO operations
   More secure

Deterministic encryption
   Encrypt('123-45-6789') = 0x85a55d3f
   Repeat: Encrypt('123-45-6789') = 0x85a55d3f
   Allows for transparent retrieval of encrypted
   data AND equality comparison
      E.g. in WHERE clauses and joins, distinct,
      group by
```

- ➔ Two types of encryption
  - ➔ Randomized encryption uses a method that encrypts data in a less predictable manner
  - ➔ Deterministic encryption always generates the same encrypted value for a given plaintext value

https://learn.microsoft.com/sql/relational-databases/security/encryption/always-encrypted-database-engine#configure-always-encrypted

Always Encrypted

- → Always Encrypted is a client-side encryption of sensitive data using keys that are never given to the database system
- → Only the application that has the encryption key can access the encrypted sensitive data
- → Minimal(*) application changes via server and client library enhancements

DelphiDay italian conference

Always Encrypted

- ➔ Always Encrypted is a client-side encryption technology in which data is automatically encrypted not only when it is written but also when it is read by an approved application
- ➔ Only the application that has the encryption key can access the encrypted sensitive data
- ➔ The key is never passed to SQL Server

Always Encrypted

- ➔ Always Encrypted may also be enabled in the DSN configuration or programmatically with the SQL_COPT_SS_COLUMN_ENCRYPTION pre-connection attribute
- ➔ The client driver needs to have access to the relevant certificate; MSSQL, ODBC, ... drivers do it for us

Demo

2

Manage encrypted columns in Delphi

3

DelphiDay italian conference

Delphi and encrypted columns

- ➔ A Delphi application that manages SQL Server encrypted columns must use parameterized query
- ➔ Enable both parameter encryption and result set encrypted column decryption is by setting the value of the ColumnEncryption connection string keyword to Enabled

Delphi and encrypted columns

- ➔ Use prepare method of the query
- ➔ You cannot use either literals or SQL local variables to INSERT, UPDATE, or compare with Always Encrypted columns, as the server has no access to the decrypted data
- ➔ Pay attention to the data type and the size of the parameters
- ➔ Pay attention to the randomized encryption type

DelphiDay italian conference

Demo

3

Summary

- ➔ Encryption is the process of obfuscating data using a key
- ➔ SQL Server provides several encryption mechanisms
- ➔ Always Encrypted is a feature designed to protect sensitive data with minimal* application changes via server and client library enhancements
- ➔ A Delphi application that manages SQL Server encrypted columns can use FireDAC connection with parameterized query

DelphiDay italian conference

Resources

- → Connect to Microsoft SQL Server (FireDAC)
- → How to manage Always Encrypted columns from a Delphi application
- → SQL Server encryption
- → Always Encrypted documentation
- → Analyze the impacts due to the possible change of COLLATE
- → Analyze possible impacts on client applications
- → Working with column master key stores
- → FireDAC and Microsoft Azure SQL Database
- → Credits to Gianluca Hotz

DelphiDay
italian conference

THANK YOU !

DelphiDay
italian conference