Advanced Technology Days

5. i 6. prosinca 2017., Plaza Event Centar

Powered by

Microsoft

13 ADVANCED TECHNOLOGY DAYS

Powered by:

SQL Server 2017 Always Encrypted

Josip Šaban, dipl. ing. računarstva, MBA

Sadržaj

- O predavaču
- Zašto bi trebali ostati do kraja?
- (Vrlo kratka) povijest razvoja sigurnosti u SQL Serveru
- Always Encrypted arhitektura
- Ograničenja Always Encrypted tehnologije
- Demo
- Zaključak i Q&A

O predavaču

ADVANCED
TECHNOLOGY
DAYS

O predavaču

- Završio računarstvo na FER-u i MBA na Cotrugli-u, trudi se doktorirati na FOI-u, a dodatno ga povezuju sa kraticama kao što su PMP, ITIL, MCT, MC* za SQL Server, a nađe se ponešto i za Project Server, Biztalk i .NET
- Veliki zaljubljenik u Courseru, završene 3 specijalizacije i 10-ak nezavisnih tečajeva
- Sa SQL Server-om od verzije 2000 do danas, osobna područja interesa su podatkovna analitika, izgradnja timova i ljudi te teorija upravljanja sustavima
- Prilično javna osoba…što znači: „Ako vas zanima više o meni, Google zna“ ☺
- Trenutno pokušava učiniti svijet boljim kao dio Erste IT-a, aktivni član Microsoft Community-a od prvih dana
- Kontakt e-mail: jsaban@erstegroup.com josipsaban@gmail.com

Zašto bi trebali ostati do kraja?

Nekoliko razloga...

- Ručak je tek nakon mog predavanja i ne znate što raditi do tada ☺
- Dio ste grupe mojih starih prijatelja koji su došli postavljati pitanja ☺
- Za koja neće biti vremena i dobiti će odgovore kasnije ☺
- Ukoliko prošla stavka ne vrijedi za vas, jer je prva sigurno točna...došli ste na zabavno predavanje o vrlo bitnoj tehnologiji, koja je postala dostupna u verziji SQL Server-a 2016
- Kako dobra marketinška rečenica, zar ne?
- A o čemu se stvarno radi?

Nekoliko razloga...

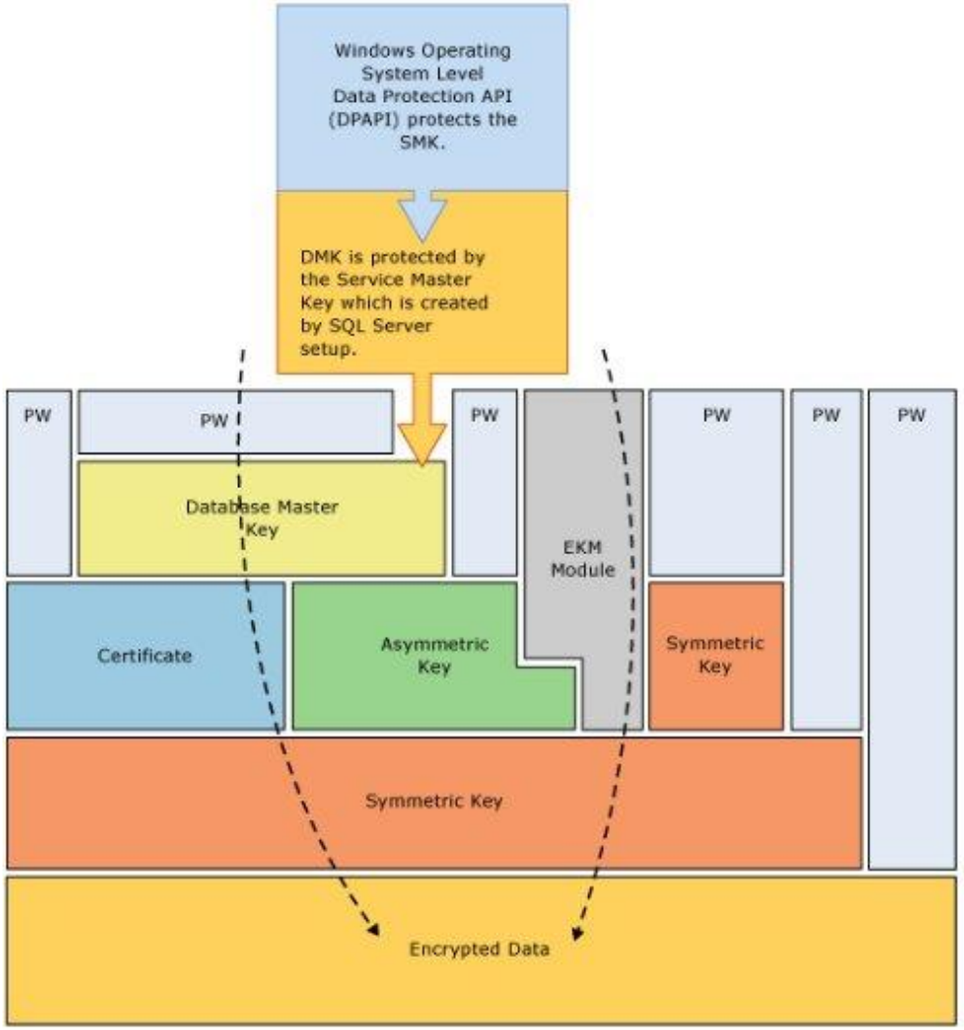
- Always Encrypted je tehnologija za sigurni prijenos podataka preko mreže, između klijenta i baze podataka, bez korištenja vanjskih biblioteka, dostupan u svim verzijama SQL Server-a (od Expressa na „više")
- Iako SSL kanal služi osiguravanju „sigurnog komunikacijskog kanala", još je bolje ukoliko ni baza podataka ni komunikacijski kanal ne znaju sadržaj podataka – kriptiranje i dekriptiranje podataka vrši se isključivo na klijentu
- Zahtijeva SQL Server 2016 ili više verzije te klijentsku aplikaciju pisanu u .NET-u 4.6 ili višem – niže verzije ili druge klijentske tehnologije nisu podržane u času izrade ove prezentacije
- Odlično rješenje za čuvanje privatnosti podataka – sve što klijent unese nije vam poznato ni dostupno, a sustav i dalje radi

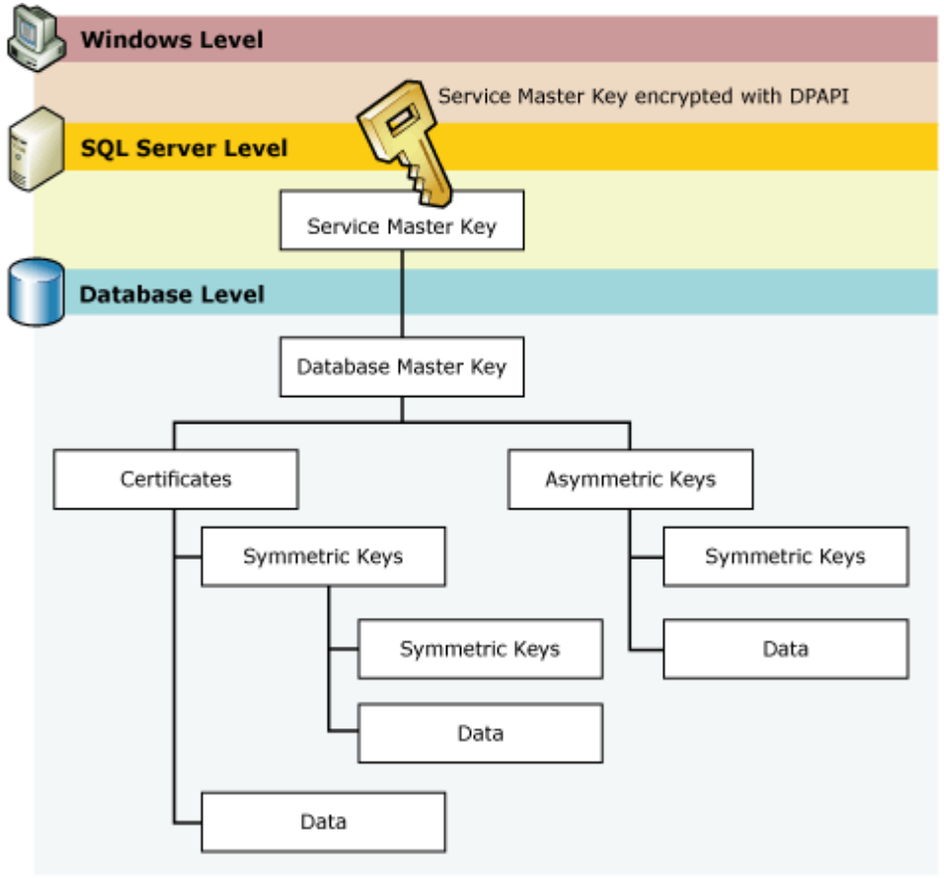
- (Vrlo kratka) povijest razvoja sigurnosti u SQL Serveru

SQL Server...od stoljeća sedmog

- SQL Server do verzije 2005 – nema ugrađenih enkripcijskih protokola
- SQL Server 2005 – simetrična/asimetrična enkripcija – dostupna kroz SQL
- SQL Server 2008 – TDE (Transparent data encryption) na nivou MDF/LDF datoteka te sigurnosnih kopije, enkripcija dostupna kroz .NET (System.Security.Cryptography)
- SQL Server 2008 R2 / 2012 - enkripcija temeljena na certifikatima
- SQL Server 2016/2017 – Always Encrypted (tema ovog predavanja) te Dynamic Data Masking
- Dynamic Data Masking je koristan za razvoj aplikacija – za prikaz podataka – ali i dalje nesiguran – DBA ili neki drugi korisnik može vidjeti podatke u bazi (123-45-6789 = XXX-XX-6789)

SQL Server...od stoljeća sedmog





• Always Encrypted arhitektura

Always Encrypted arhitektura

- Zašto kriptirati – sigurnost, regulatorna podrška, PII standardi („Personally Identifiable Information") – tko može spriječiti DBA korisnike u pregledu podataka?
- Performanse – kriptiranje i dekriptiranje rade se na klijentu/srednjem sloju
- Sigurnost – podaci su jedino vidljivi ukoliko postoji certifikat
- https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine
- Podaci se nikad ne nalaze u nekriptiranom stanju, bilo da su neaktivni (nisu u transakcijama ili dohvatima) ili dok su aktivni (uključujući i dok su u memoriji)

Always Encrypted arhitektura

- Always Encrypted sa TDE-om za kompletno kriptografsko rješenje
- Always Encrypted zatvara „rupu" u kriptografskim rješenjima na SQL Server platformi, osobito u slučaju regulatornih zahtjeva – administratori baze podataka nemaju pristup podacima
- Zahtjevi:

- SQL Server 2016 SP1 ili viši (od SP1 podržane sve verzije, uključujući Express)
- .NET 4.6 ili viši, mora se koristiti ADO.NET za pristup bazi podataka
- Certificate store – koristi se za spremanje Master ključa
- Kriptirani tekstualni podaci moraju koristiti BINARY2 collation LATIN1_GENERAL_BIN21

Always Encrypted arhitektura

ADVANCED TECHNOLOGY DAYS

- Always Encrypted podržava dvije vrste enkripcije:

- Slučajnu enkripciju – iste vrijednosti će imati različite kriptirane vrijednosti – sigurnije ali ne podržava usporedbe, JOIN operacije, grupiranje, pretraživanje te indeksiranje

  (225) 555-1234 = 0x0003456

  (225) 555-1234 = 0x00078910

- Deterističku enkripciju – iste vrijednosti će imati iste kriptirane vrijednosti – manje sigurno ali nema gornja ograničenja – podložno „pogađanju" ako se radi o kolonama čiji je sadržaj generalno poznat („Spol", „Država", „Grad", …)

  (225) 555-1234 = 0x0003456

  (225) 555-1234 = 0x0003456

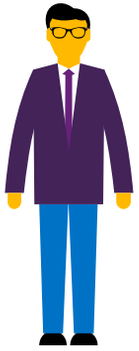
Always Encrypted arhitektura

- Always Encrypted koristi dvije vrste ključeva:
- Column Encryption Keys (CEK)
- Column Master Keys (CMK)

- CEK ključevi se koriste za enkripciju CMK ključeva

- CMK ključevi služe za enkripciju jedne kolone u tablici
- Na svaki CMK se mogu vezati dva CEK-a što omogućava rotaciju CEK-ova (npr. u slučaju isticanja validnosti ključa)
- CMK se mora instalirati na svaki klijentski stoj koji treba pristup nekriptiranim podacima – ti ključevi se NE instaliraju na računalo gdje se nalazi SQL Server

Always Encrypted arhitektura



1. Generate CEKs and Master Key

Column Encryption Key (CEK)

Column Master Key (CMK)

2. Encrypt CEK

Encrypted CEK

3. Store Master Key Securely

CMK Store:

- *Certificate Store*
- *HSM*
- *Azure Key Vault*
- *…*

CMK

4. Upload Encrypted CEK to DB

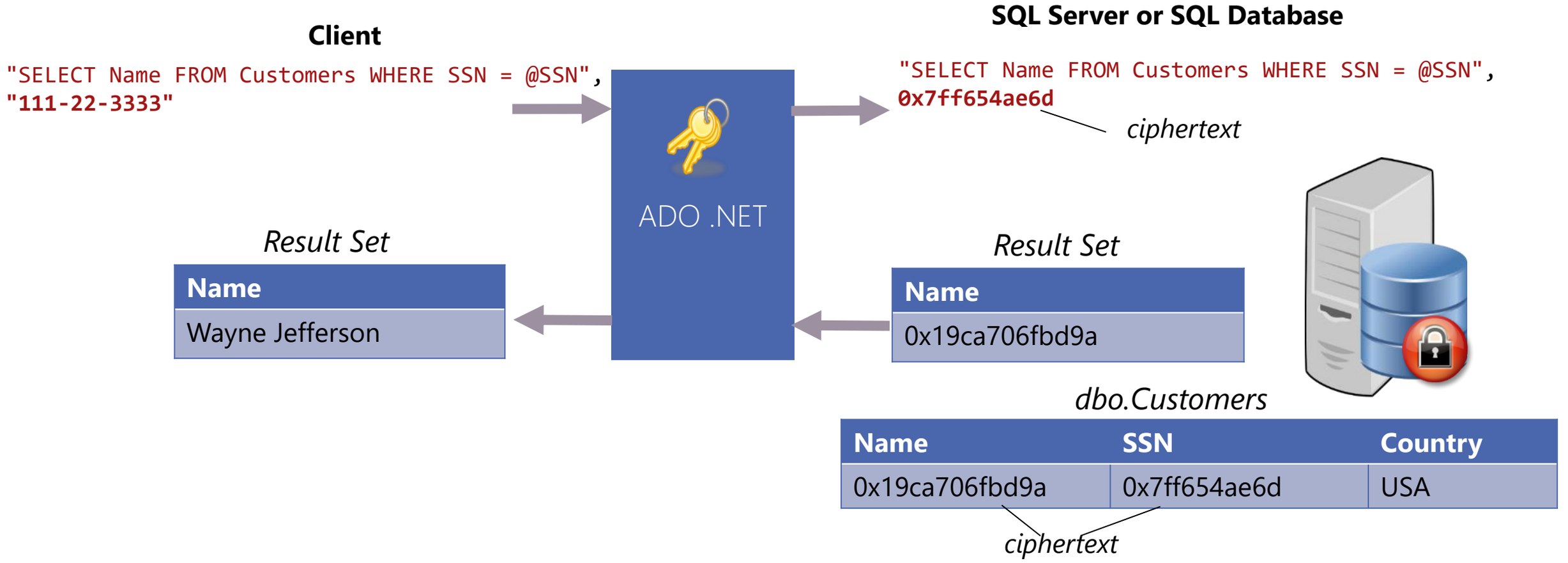
Encrypted CEK

Database

Always Encrypted arhitektura - MSDN

Encrypted sensitive data and corresponding keys are never seen in plaintext in SQL Server

ADVANCED TECHNOLOGY DAYS



Ograničenja Always Encrypted tehnologije

ADVANCED TECHNOLOGY DAYS

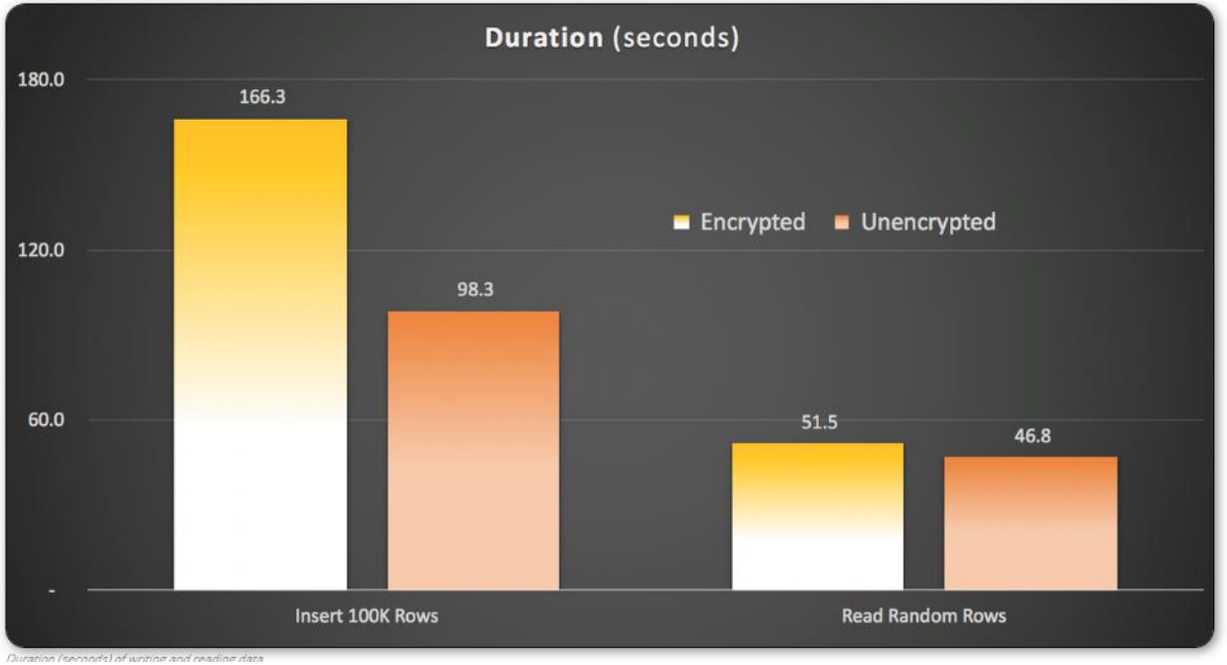
Ograničenja Always Encrypted tehnologije

- Ne možete koristiti sa slijedećim tipovima podataka:
- XML
- ROWVERSIONTIMESTAMP
- IMAGE
- TEXTNTEXT
- SQL_VARIANT
- HIERARCHYID
- GEOGRAPHY
- GEOMETRY

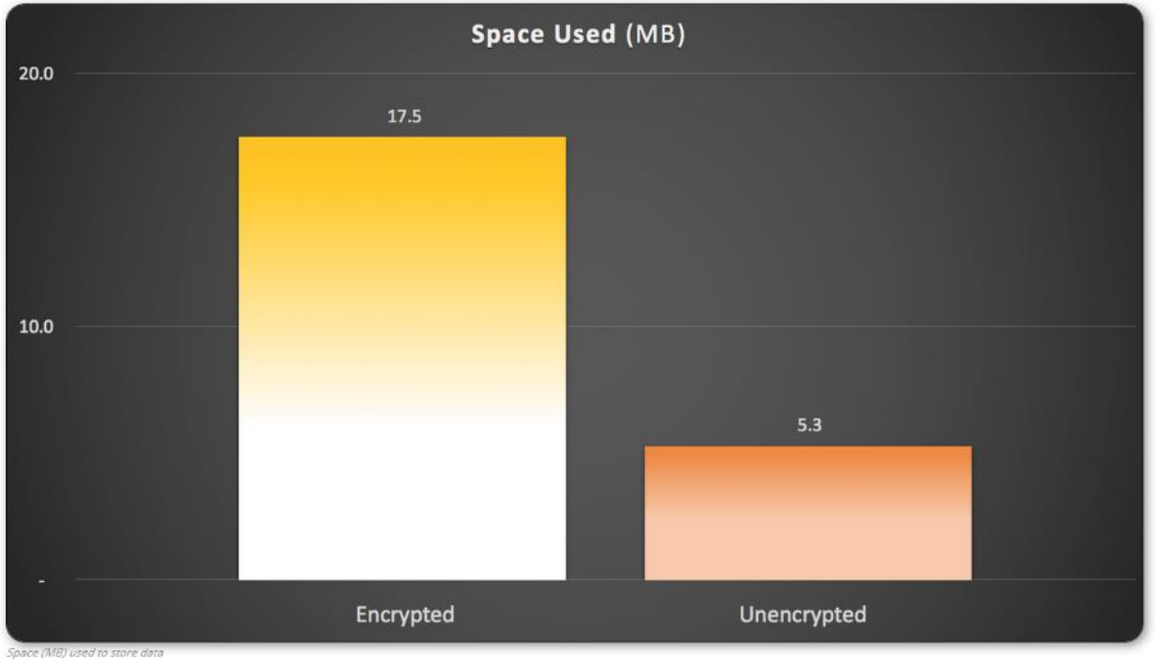
- Ostala ograničenja (vrijede za verziju SQL Server-a 2017)
- https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine#feature-details

Utjecaj na performanse

- https://sqlperformance.com/2015/08/sql-server-2016/perf-impact-always-encrypted



Duration (seconds) of writing and reading data



Space (MB) used to store data

Demo

Demo - ukratko

- Aplikacija radi kriptiranje – šalje čist tekst ADO.NET driveru, nakon čega se kriptirani podaci šalju prema bazi
- Jedina promjena u .NET aplikaciji je promjena connection stringa, koja ukazuje da je uključena enkripcija
- Nakon toga se ADO.NET brine o kriptiranju i de-kriptiranju podataka
- Cilj – onemogućiti administratorskim korisnicima pristup dekriptiranim podacima

Demo - ukratko

- Dva ključa su osnova cijelog procesa – Column Master Key (na klijentskom stroju, u key store-u) koji služi zaštiti Colum Encryption Key-a
- Ovakva arhitektura onemogućava dekriptiranje podataka SQL Server-u
- Drugi ključ (Column Encryption key) se sprema na SQL Server, te služi za kriptiranje i dekriptiranje Always Encrypted kolona u bazi

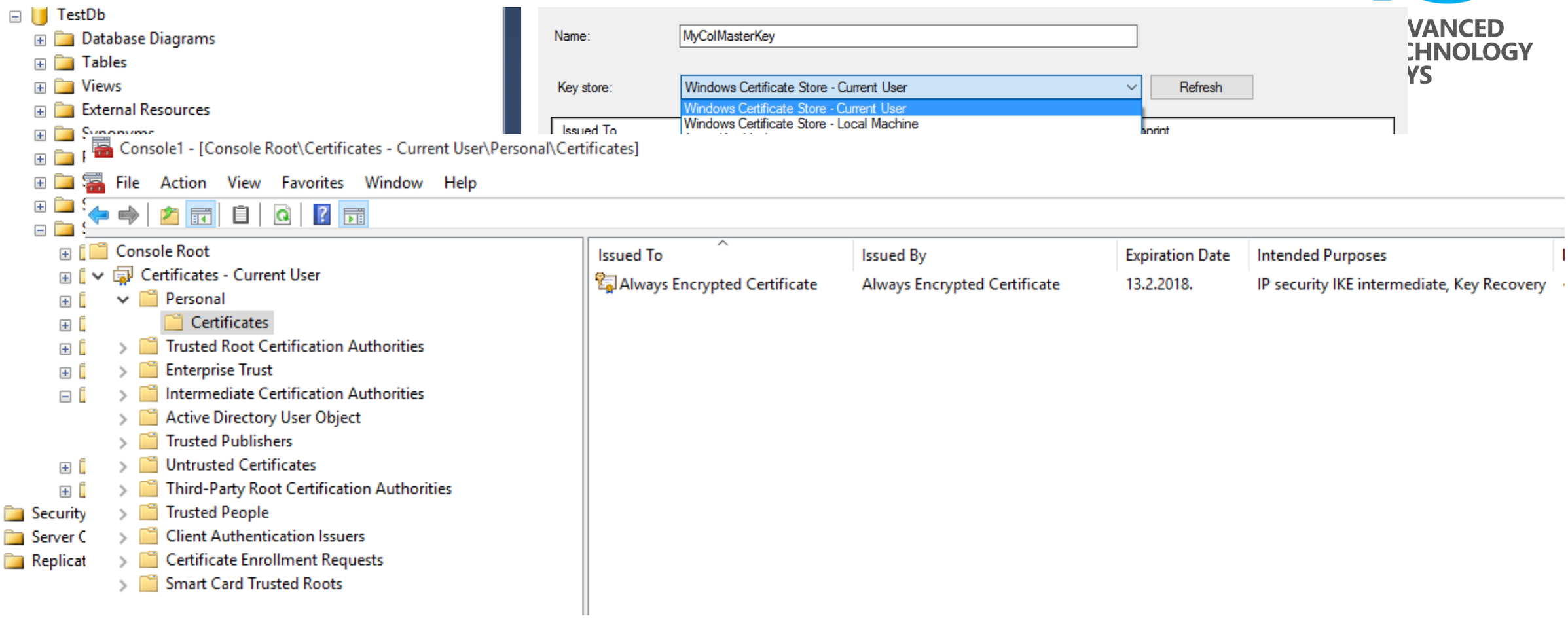
Demo - ukratko

- Jednom kad je ADO.NET dekriptirao Column Encryption key, korištenjem Column Master key-a, može ga iskoristiti za kriptiranje i dekriptiranje Always Encrypted kolona

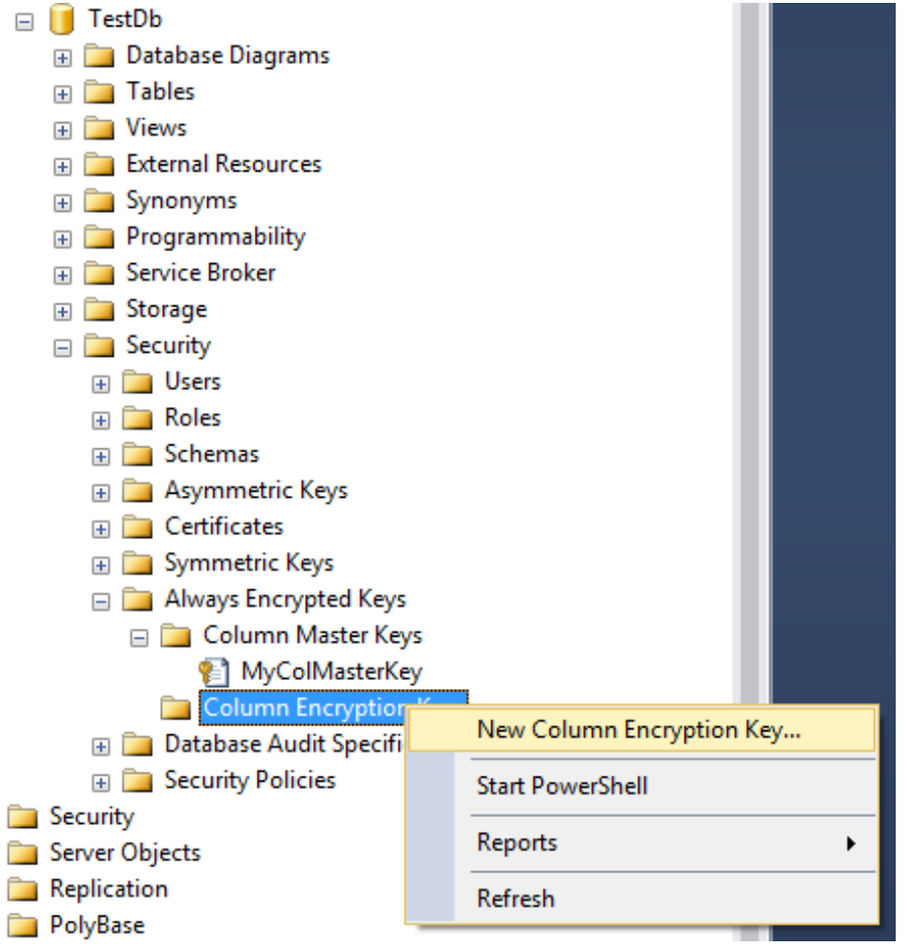
- Za postavljanje arhitekture potrebno je:

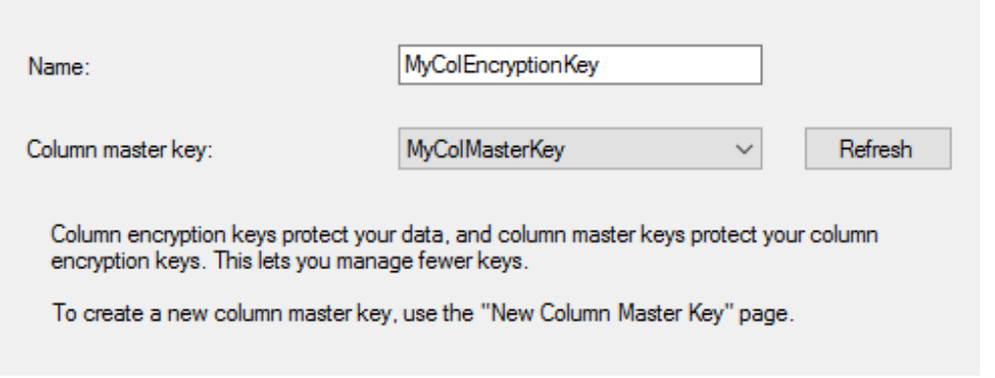
- Aplikacija koja koristi .NET 4.6 framework ili viši
- Instanca SQL Server-a 2016 ili višeg (potrebna je verzija 2016 SP1 za korištenje i Express verzija)
- Certificate store – u koji spremamo Column Master Key
- Column Master Key
- Column Encryption Key
- Tablica u bazi sa Always Encrypted kolonama

Column Master Key



Column Encryption Key





Stvaramo demo tablicu


```
CREATE TABLE dbo.MojDemo
(
        ID INT IDENTITY(1,1) PRIMARY KEY,
        StringOne NVARCHAR(255),
        StringTwo NVARCHAR(255),
        NekiBitanDatum DATE ENCRYPTED WITH ( ENCRYPTION_TYPE = RANDOMIZED, ALGORITHM =
                'AEAD_AES_256_CBC_HMAC_SHA_256', COLUMN_ENCRYPTION_KEY = MyColEncryptionKey ),
        NekiBitanString NVARCHAR(50) COLLATE Latin1_General_BIN2 ENCRYPTED WITH ( ENCRYPTION_TYPE =
                DETERMINISTIC, ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256',
                COLUMN_ENCRYPTION_KEY = MyColEncryptionKey )
);
```

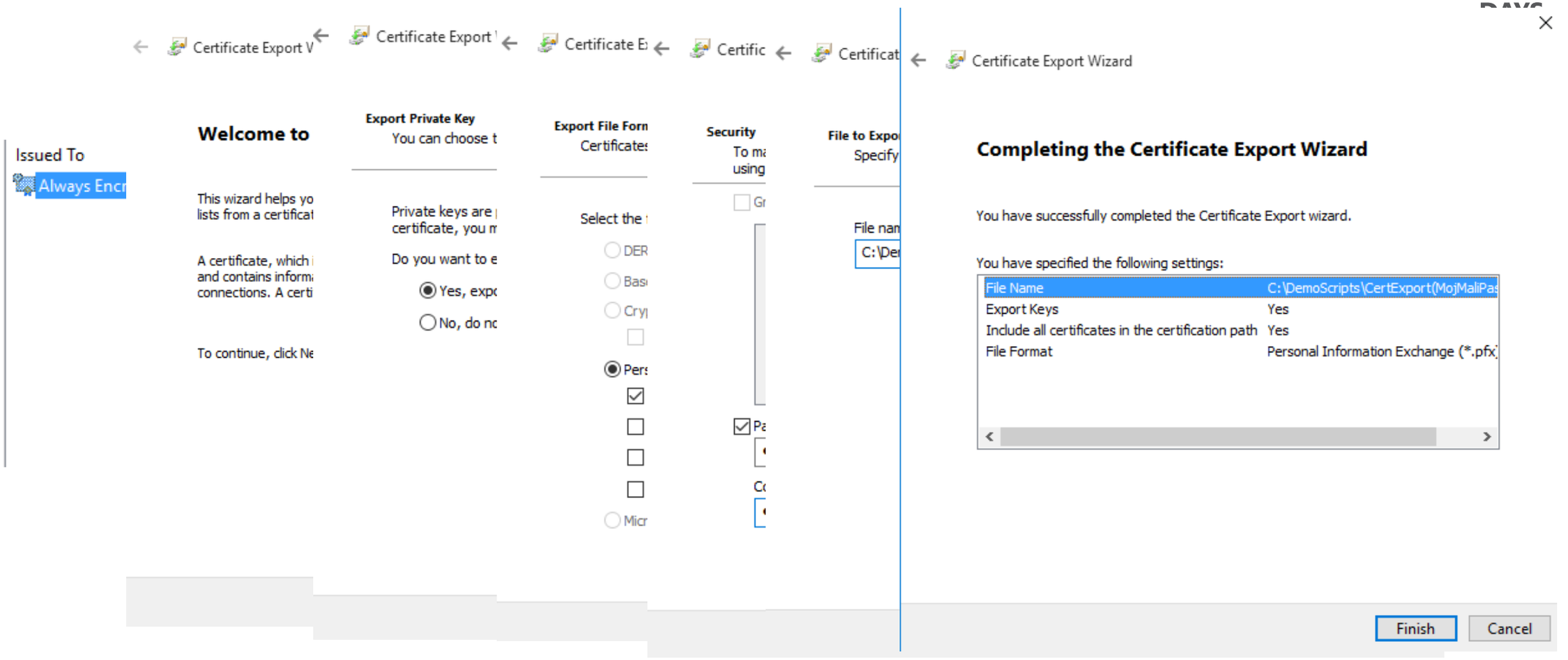
Procedura za unos


```
CREATE PROCEDURE AE_Insert ( @StringOne NVARCHAR(255), @StringTwo NVARCHAR(255), @NekiBitanDatum DATE,
@NekiBitanString NVARCHAR(50) ) AS
        INSERT INTO dbo.MojMaliDemo (StringOne, StringTwo, NekiBitanDatum, NekiBitanString)
                VALUES (@StringOne,@StringTwo,@NekiBitanDatum,@NekiBitanString);

-- Ne ide standardno :( - nije poslano ADO.NET enkriptirano
EXEC AE_Insert @StringOne = 'AAAAA', @StringTwo = 'BBBBB',
        @NekiBitanDatum = '2016-01-01', @NekiBitanString = 'MojBitanString';
```

Msg 206, Level 16, State 2, Procedure AE_Insert, Line 0 [Batch Start Line 27] Operand type clash: varchar is incompatible with date encrypted with (encryption_type = 'RANDOMIZED', encryption_algorithm_name = 'AEAD_AES_256_CBC_HMAC_SHA_256', column_encryption_key_name = 'MyColEncryptionKey', column_encryption_key_database_name = 'TestDb')

Uvoz/izvoz certifikata



C# kod

- Programski kod je u dodatku ove prezentacije
- Nakon izvršenja koda treba provjeriti
- Koje podatke vidi trenutno ulogirani korisnik?
- Koje podatke vidi neki drugi, administratorski, korisnik?

```
SELECT *  FROM dbo.MojDemo
```

- Pitanje za dodatne bodove – zašto „ovo" vide i svi lokalni korisnici?

A što ako su klijent i baza na istom serveru?

- „If you ran your SSMS from the same box as you used to run your C# app (that does have the CMK in its certificate store) this behavior should be by design - it just means the SSMS itself can use ADO.NET for encrypting/decrypting data. As far as I understand DBAs are NOT meant to use the same computers as DB users - only in this case Always Encrypted can work as expected.”

- Ovo ponašanje se mijenja kroz verzije i treba provjeriti kako se SSMS ponaša sa verzijom koju vi koristite

- Zaključak i Q&A

13
ADVANCED
TECHNOLOGY
DAYS

Zaključak i Q&A

- Always Encrypted samo traži promjenu connection string-a
- Podaci su zaštićeni u trenutku napuštanja klijentskog računala
- Kriptirani podaci su zaštićeni od administratora servera i baze podataka
- Upravljanje certifikatima ključno je za zaštitu podataka

ADVANCED
TECHNOLOGY
DAYS

Powered by

Microsoft