CHƯƠNG 4
DATABASE SECURITY

10/2/2021

ThS.Nguyễn Duy
duyn@uit.edu.vn

Content

- What is Database Security?
- What is Database Security Technical?
- How to deploy DBF?

Content

- **What is Database Security?**
- What is Database Security Technical?
- How to deploy DBF?

What is Database Security?

- There are four key issues in the security of databases just as with all security systems:
  - Confidentiality
  - Integrity
  - Authenticity
  - Availability

Confidentiality

- Need to ensure that **confidential data** is only available to correct people
- Need to ensure that entire database is **security from external and internal system breaches**
- Need to provide for **reporting** on who has accessed what data and what they have done with it
- Mission critical and Legal sensitive data must be highly security at **the potential risk of lost business and litigation**

Integrity

- Need to verify that any external data has the correct formatting and other metadata
- Need to verify that all input data is accurate and verifiable
- Need to ensure that data is following the correct work flow rules for your institution/corporation
- Need to be able to report on all data changes and who authored them to ensure compliance with corporate rules and privacy laws.

Authenticity

- Need to ensure that the data has been edited by an authorized source
- Need to confirm that users accessing the system are who they say they are
- Need to verify that any outbound data is going to the expected receiver
- Need to verify that all report requests are from authorized users

Availability

- Data needs to be available at all necessary times
- Data needs to be available to only the appropriate users
- Need to be able to track who has access to and who has accessed what data

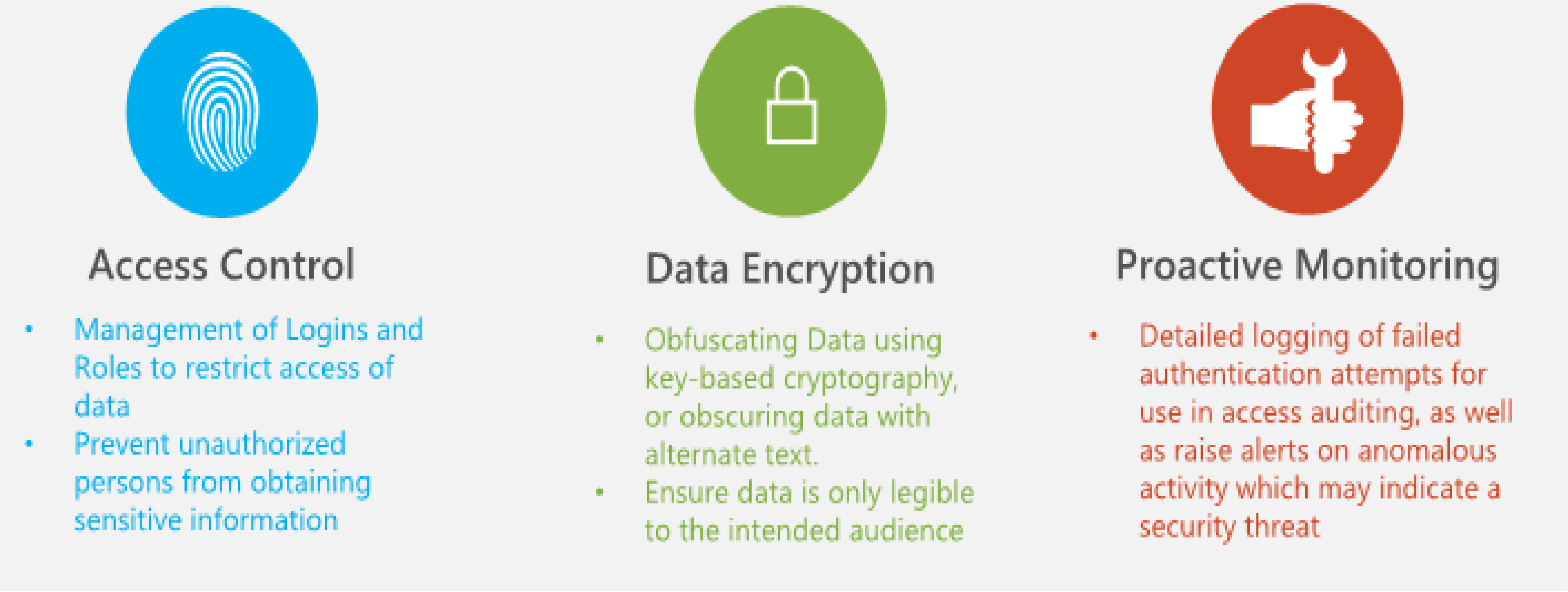
Security for Database

- Design Database: architecture, encrypt,….
- Security functions of Database: Oracle, Microsoft,…
- 3rd security option: Database Firewall,…

Content

- What is Database Security?
- **What is Database Security Technical?**
- How to deploy DBF?

Technical Solution



Access Control

- Management of Logins and Roles to restrict access of data
- Prevent unauthorized persons from obtaining sensitive information

Data Encryption

- Obfuscating Data using key-based cryptography, or obscuring data with alternate text.
- Ensure data is only legible to the intended audience

Proactive Monitoring

- Detailed logging of failed authentication attempts for use in access auditing, as well as raise alerts on anomalous activity which may indicate a security threat

Access Control

Protect your organization, data and people

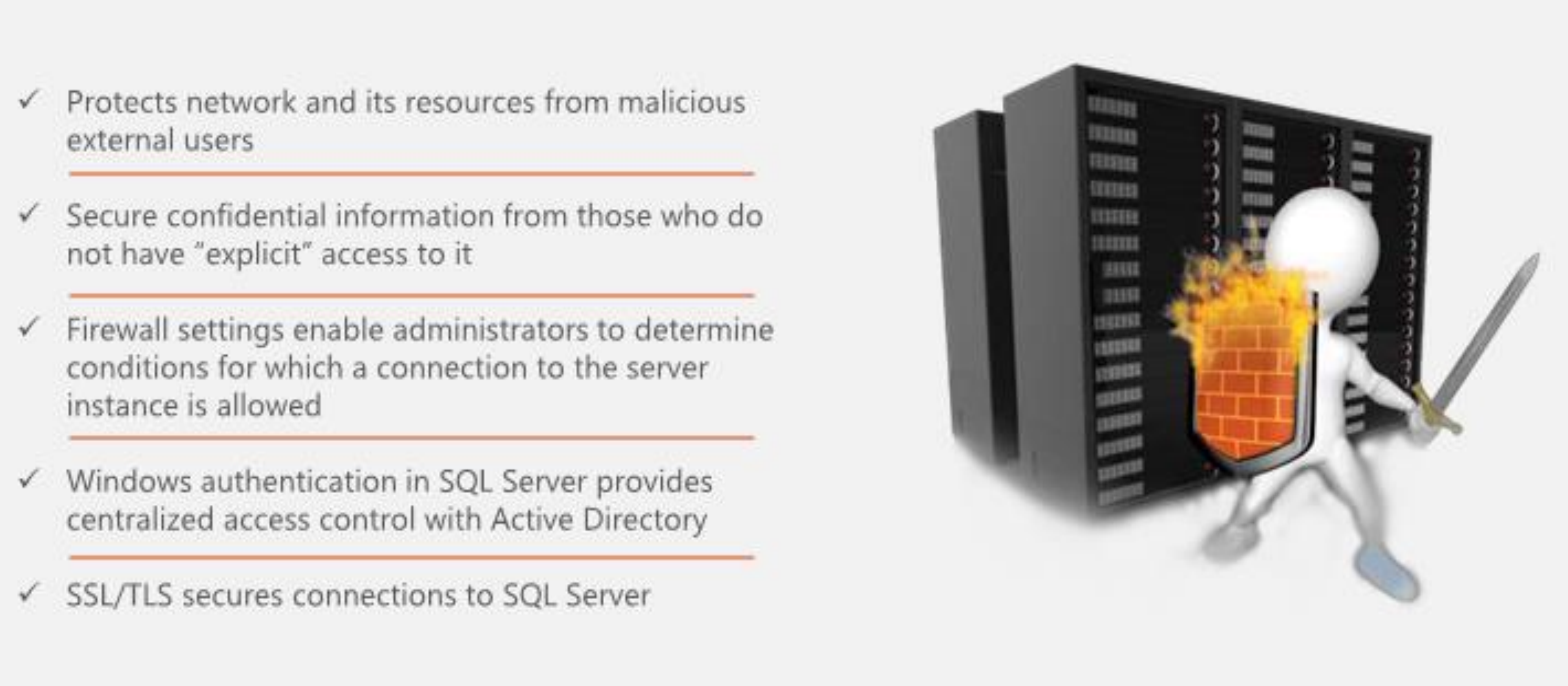
Access Control

Identification – Are you allowed?

Authentication – Who are you?

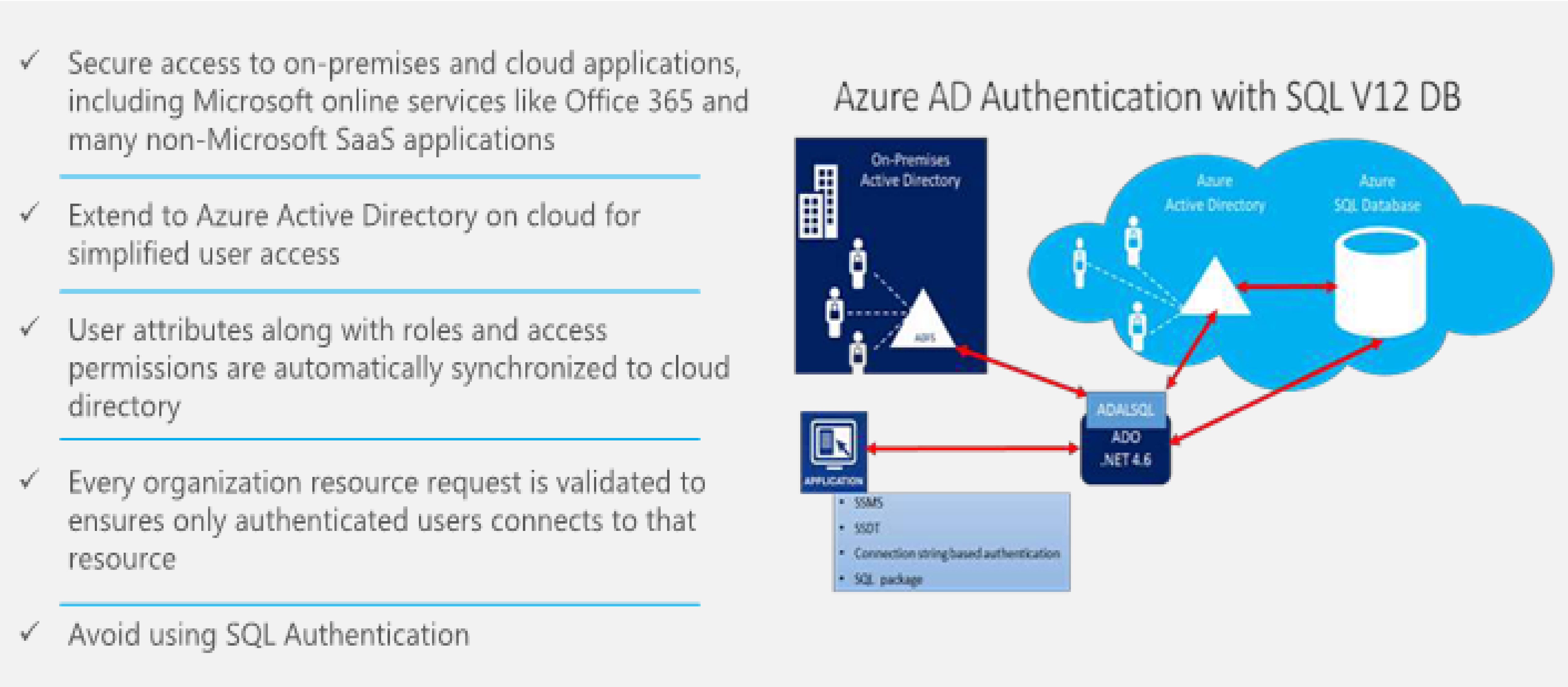
Authorization – What all could you do?

Firewall



- ✓ Protects network and its resources from malicious external users
- ✓ Secure confidential information from those who do not have “explicit” access to it
- ✓ Firewall settings enable administrators to determine conditions for which a connection to the server instance is allowed
- ✓ Windows authentication in SQL Server provides centralized access control with Active Directory
- ✓ SSL/TLS secures connections to SQL Server

AD Authentication

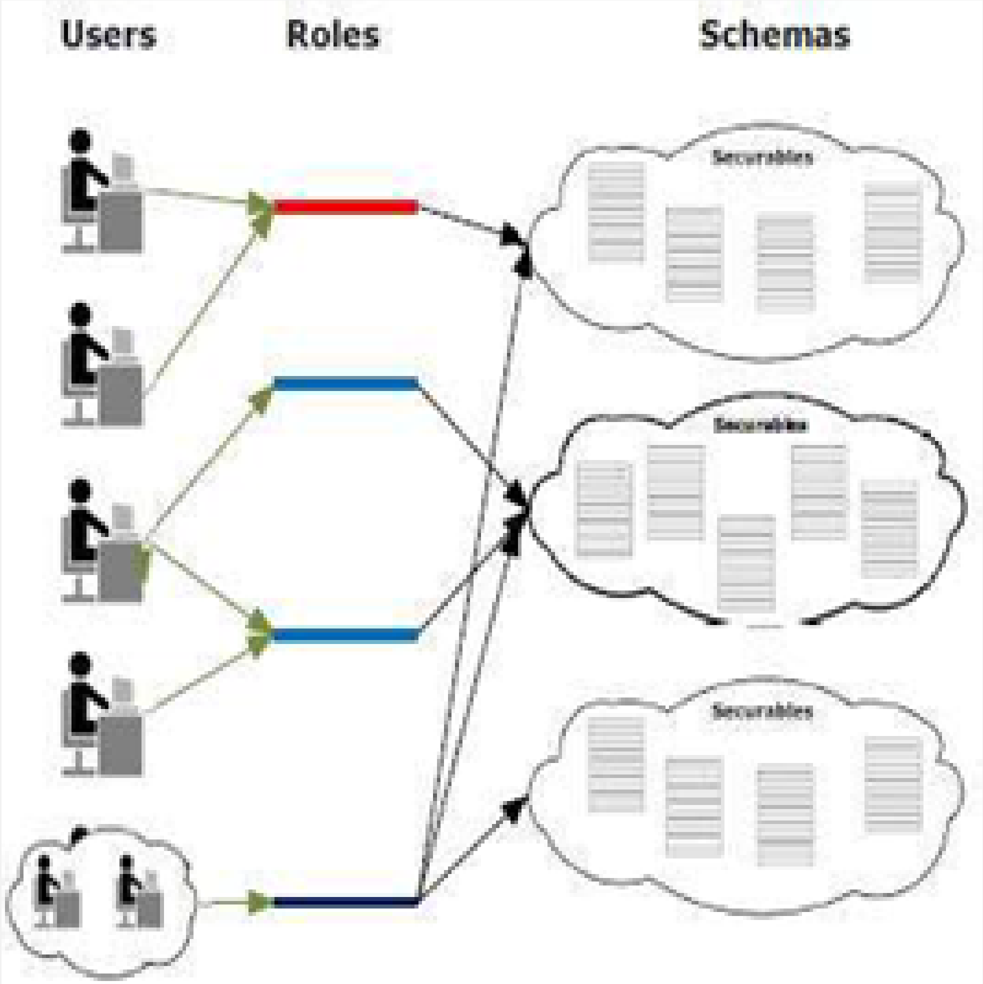


- ✓ Secure access to on-premises and cloud applications, including Microsoft online services like Office 365 and many non-Microsoft SaaS applications
- ✓ Extend to Azure Active Directory on cloud for simplified user access
- ✓ User attributes along with roles and access permissions are automatically synchronized to cloud directory
- ✓ Every organization resource request is validated to ensures only authenticated users connects to that resource
- ✓ Avoid using SQL Authentication

Azure AD Authentication with SQL V12 DB

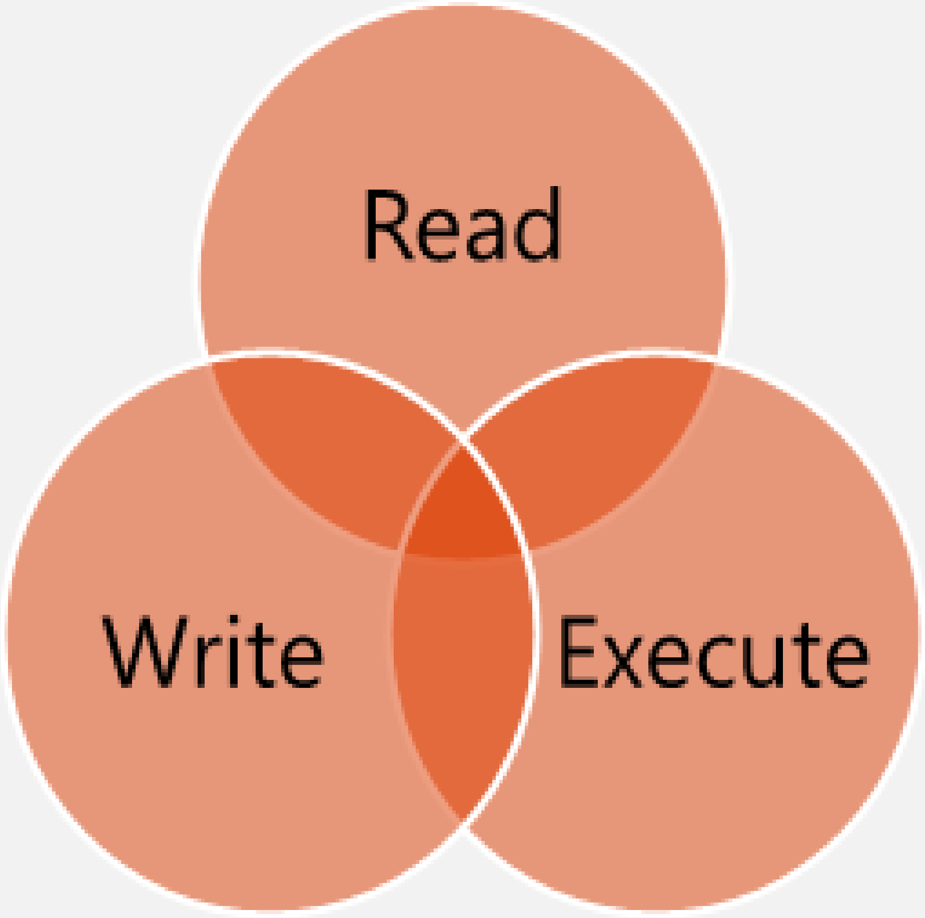
Separation of Roles

- ✓ Not every authenticated user should access everything. Only authorized users should get access to any resource/data
- ✓ Role-based access control (RBAC) is an approach to restricting system access to authorized users.
- ✓ Permissions are associated with roles, and users are assigned to appropriate roles
- ✓ Roles are created for the various job functions in an organization and users are assigned roles based on their responsibilities and qualifications
- ✓ Users can be easily reassigned from one role to another

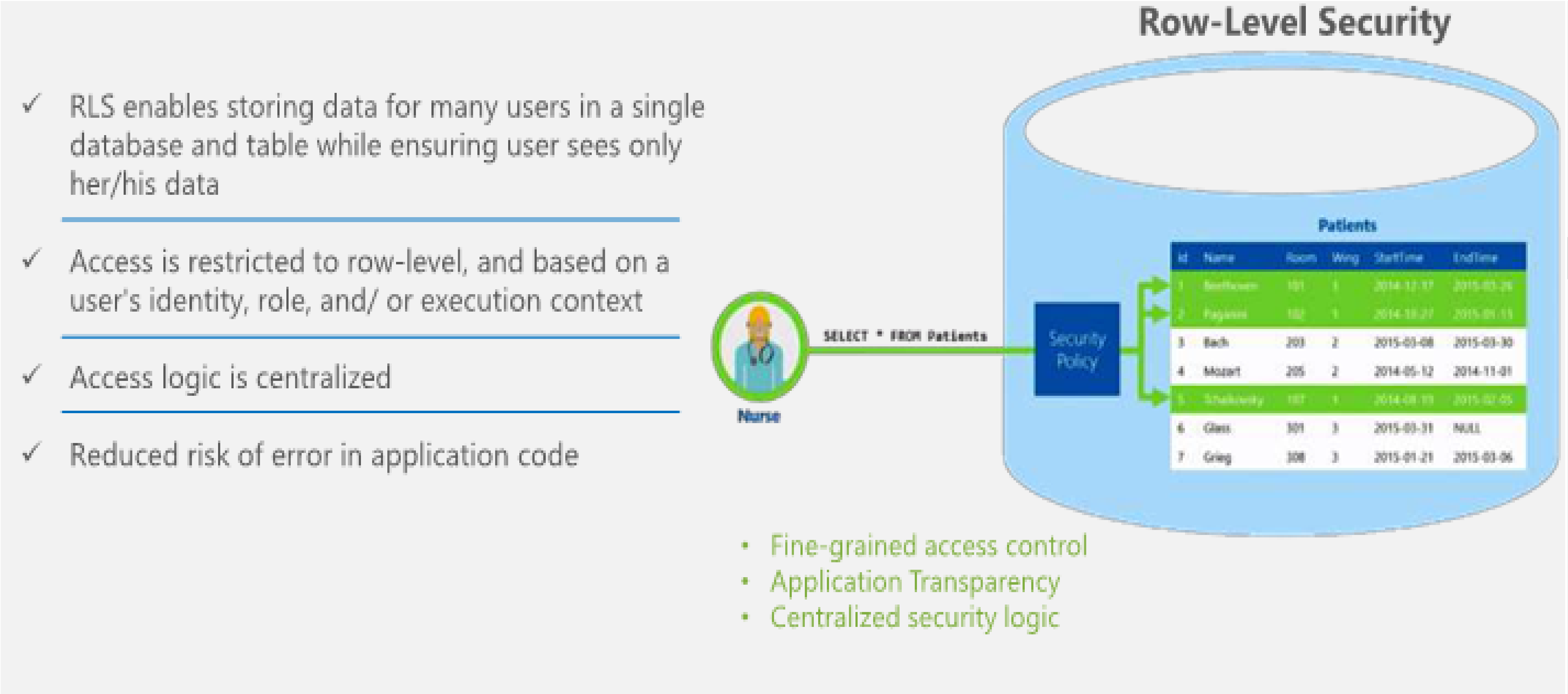


Permission

- ✓ Granular access permissions for the organization's repositories
- ✓ Admin must ensure that minimum required permissions are given to any role/user to allow it complete the required tasks. **No less and No More**
- ✓ Read, Write and Execute - Ensure right user have right set of permissions, to avoid any malicious or accidental threat to data security
- ✓ Regular audit of permissions must be done



Row-Level Security



Row-Level Security

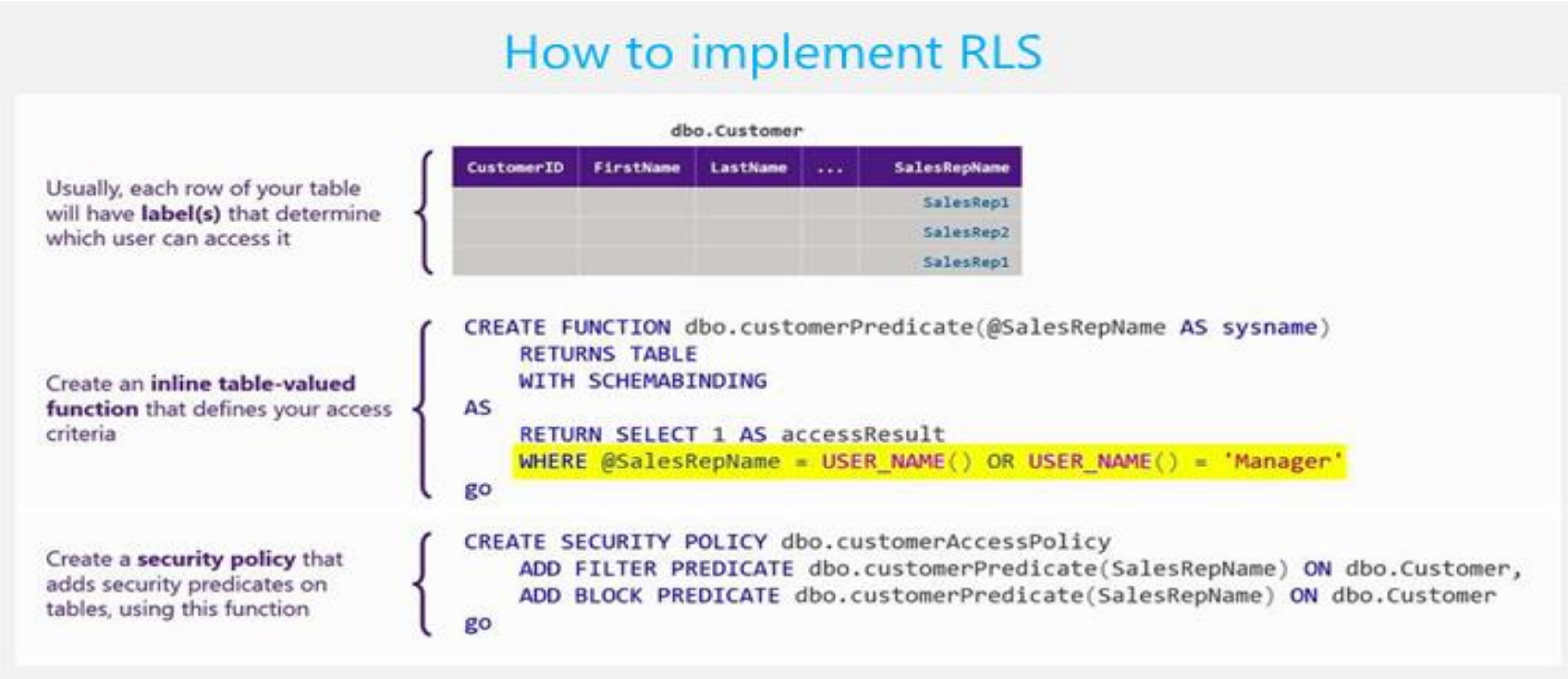
- ✓ RLS enables storing data for many users in a single database and table while ensuring user sees only her/his data
- ✓ Access is restricted to row-level, and based on a user's identity, role, and/ or execution context
- ✓ Access logic is centralized
- ✓ Reduced risk of error in application code

Patients

Id	Name	Room	Wing	StartTime	EndTime
1	Beethoven	101	1	2014-12-17	2015-03-26
2	Paganini	102	1	2014-10-27	2015-01-11
3	Bach	203	2	2015-03-08	2015-03-30
4	Mozart	205	2	2014-05-12	2014-11-01
5	Tchaikovsky	107	1	2014-08-15	2015-02-05
6	Glass	301	3	2015-03-31	NULL
7	Grieg	308	3	2015-01-21	2015-03-06

- Fine-grained access control
- Application Transparency
- Centralized security logic

Row-Level Security



How to implement RLS

Usually, each row of your table will have **label(s)** that determine which user can access it

dbo.Customer

CustomerID	FirstName	LastName	...	SalesRepName
				SalesRep1
				SalesRep2
				SalesRep1

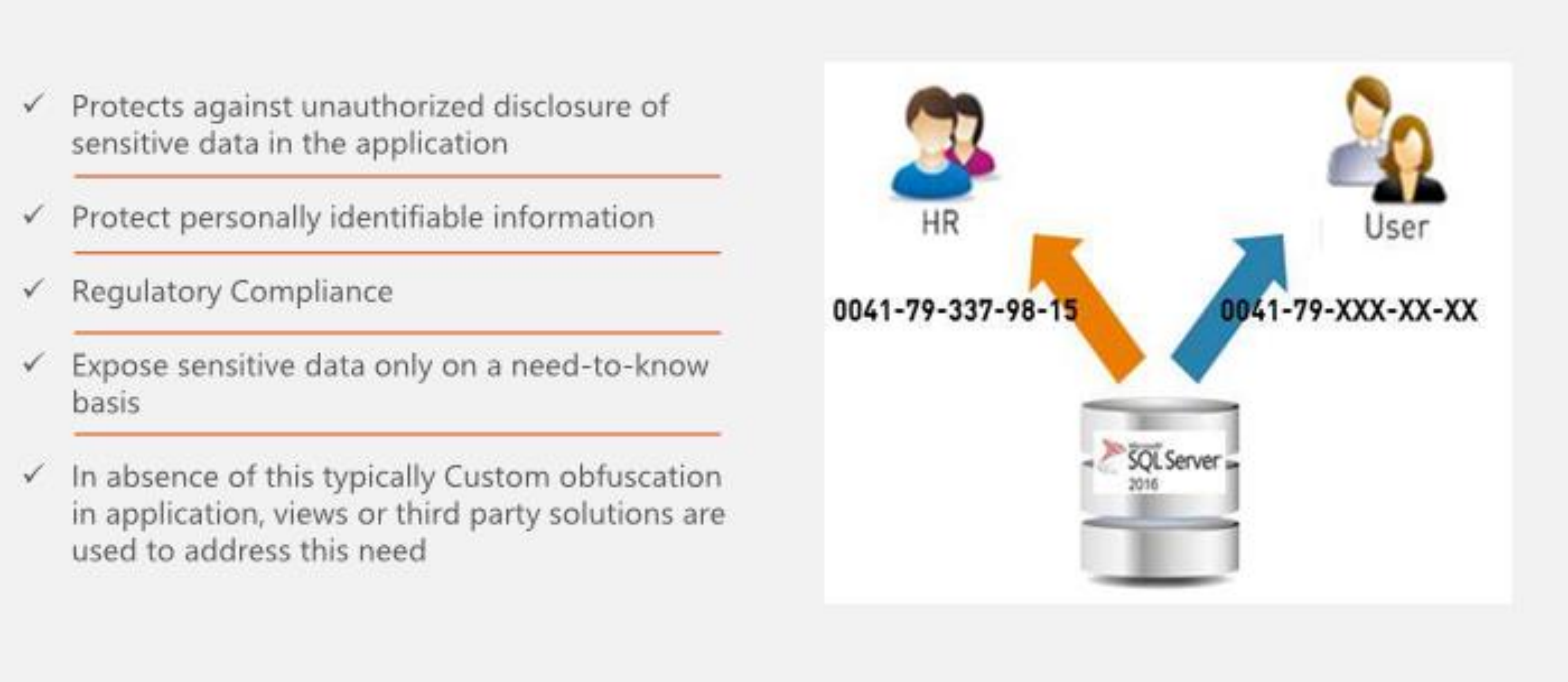
Create an **inline table-valued function** that defines your access criteria

```
CREATE FUNCTION dbo.customerPredicate(@SalesRepName AS sysname)
    RETURNS TABLE
    WITH SCHEMABINDING
AS
    RETURN SELECT 1 AS accessResult
    WHERE @SalesRepName = USER_NAME() OR USER_NAME() = 'Manager'
go
```

Create a **security policy** that adds security predicates on tables, using this function

```
CREATE SECURITY POLICY dbo.customerAccessPolicy
    ADD FILTER PREDICATE dbo.customerPredicate(SalesRepName) ON dbo.Customer,
    ADD BLOCK PREDICATE dbo.customerPredicate(SalesRepName) ON dbo.Customer
go
```

Dynamic Data Masking



- ✓ Protects against unauthorized disclosure of sensitive data in the application
- ✓ Protect personally identifiable information
- ✓ Regulatory Compliance
- ✓ Expose sensitive data only on a need-to-know basis
- ✓ In absence of this typically Custom obfuscation in application, views or third party solutions are used to address this need

ENCRYPTION

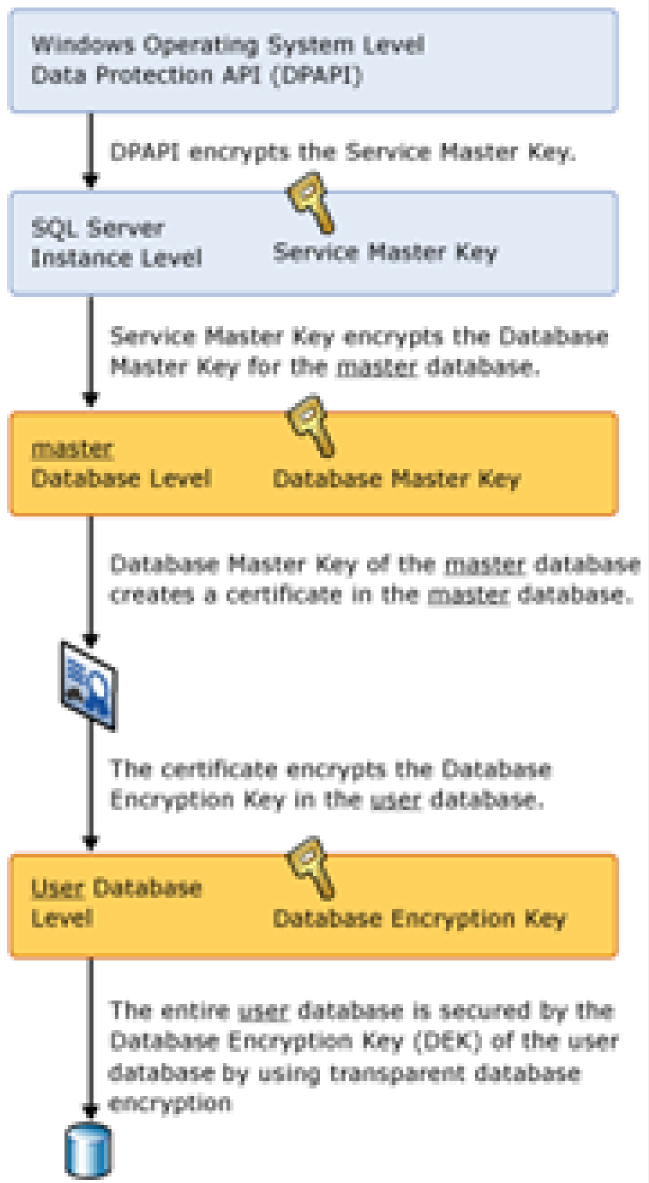
Encryption - Transparent Data Encryption (TDE)

Data protected "at rest"

Encryption/Decryption is transparent to application - no changes to code required

Does not require schema modification during implementation

Azure Services auto-manages server certificates and encryptions keys - *rotates every 90 days by Microsoft*



DPAPI encrypts the Service Master Key.

Service Master Key encrypts the Database Master Key for the master database.

Database Master Key of the master database creates a certificate in the master database.

The certificate encrypts the Database Encryption Key in the user database.

The entire user database is secured by the Database Encryption Key (DEK) of the user database by using transparent database encryption

Encryption – The need for Always Encrypted

Data disclosure prevention

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

Queries on encrypted data

Support for equality comparison, including join, group by, and distinct operators

Application transparency

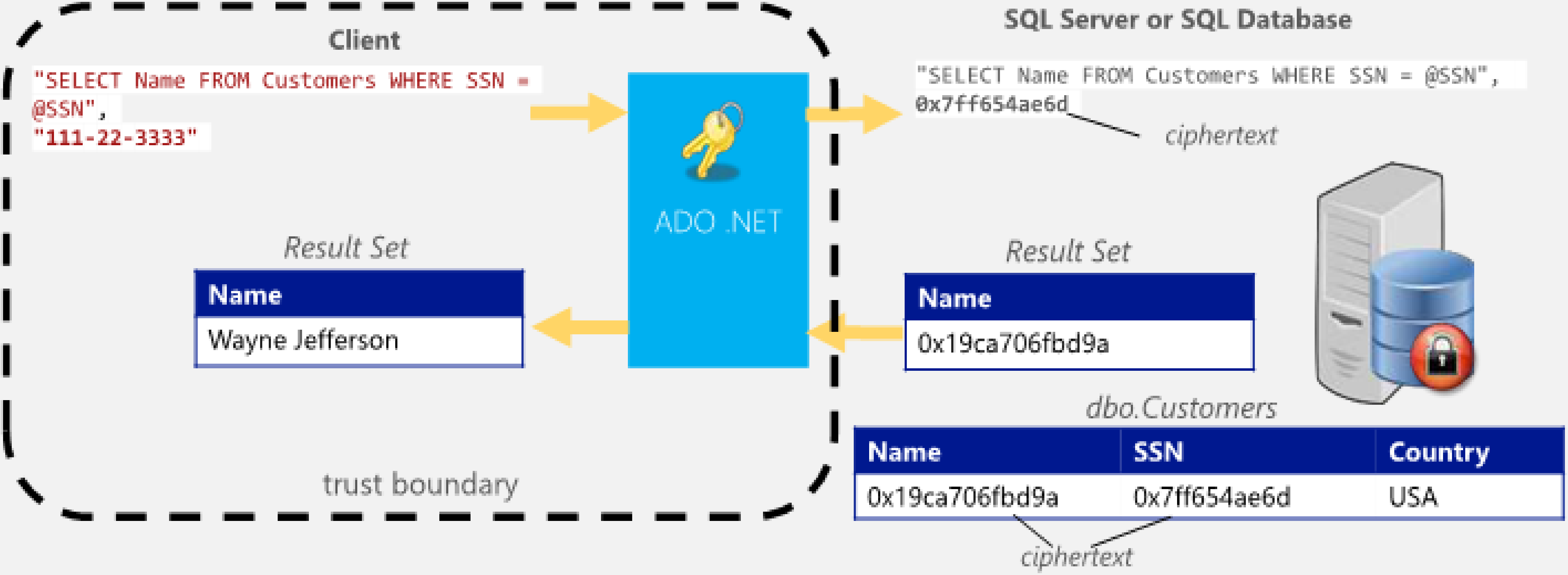
Minimal application changes via server and client library enhancements

Allows customers to securely store sensitive data outside of their trust boundary.
Data remains protected from high-privileged, yet unauthorized, users.

Encryption – How it Works

Help protect data at rest and in motion, on-premises & cloud

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



Client

```
"SELECT Name FROM Customers WHERE SSN =
@SSN",
"111-22-3333"
```

ADO .NET

Result Set

Name
Wayne Jefferson

trust boundary

SQL Server or SQL Database

```
"SELECT Name FROM Customers WHERE SSN = @SSN",
0x7ff654ae6d
```

ciphertext

Result Set

Name
0x19ca706fbd9a

dbo.Customers

Name	SSN	Country
0x19ca706fbd9a	0x7ff654ae6d	USA

ciphertext

Encryption - Types of encryption

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**

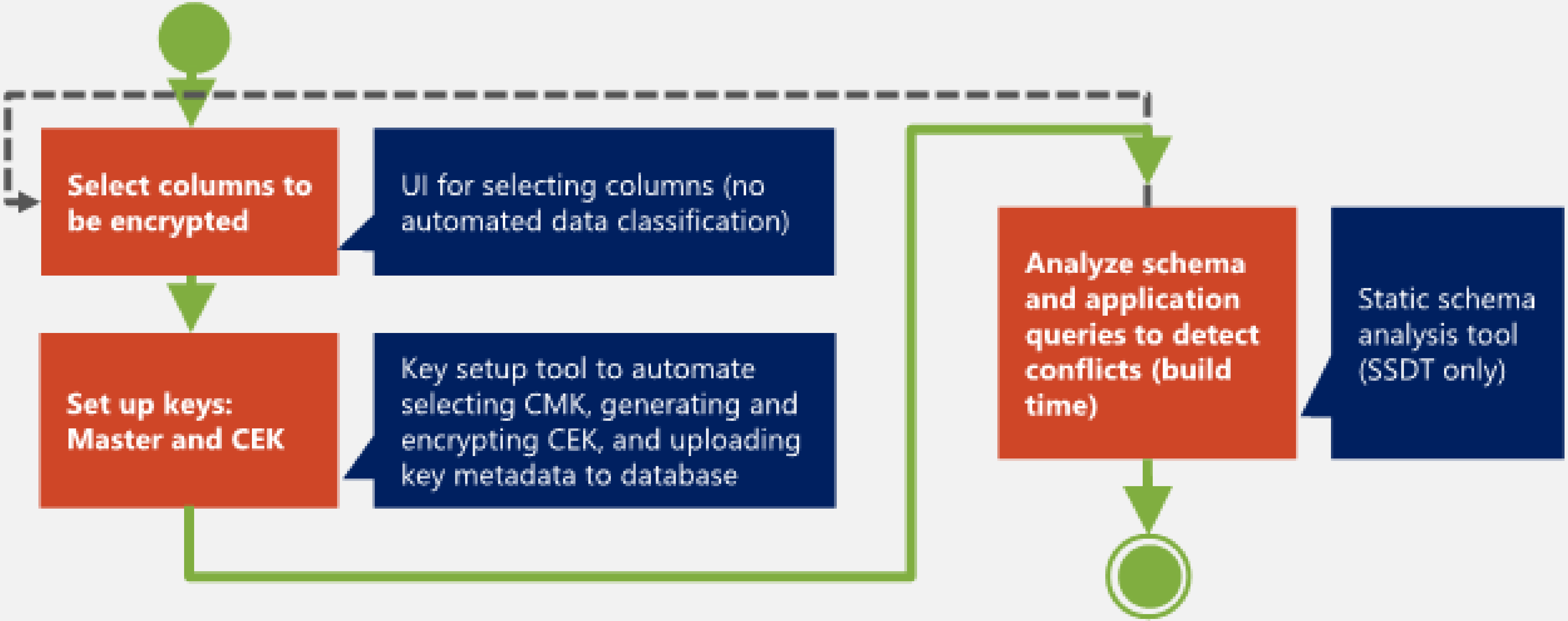
Allows for transparent retrieval of encrypted data AND equality comparison

E.g. in WHERE clauses and joins, distinct, group by

Types of encryption

- ✓ Randomized encryption uses a method that encrypts data in a less predictable manner
- ✓ Deterministic encryption uses a method that always generates the same encrypted value for any given plaintext value

Encryption - Always Encrypted Setup (SSMS or SSDT)



Select columns to be encrypted

UI for selecting columns (no automated data classification)

Set up keys: Master and CEK

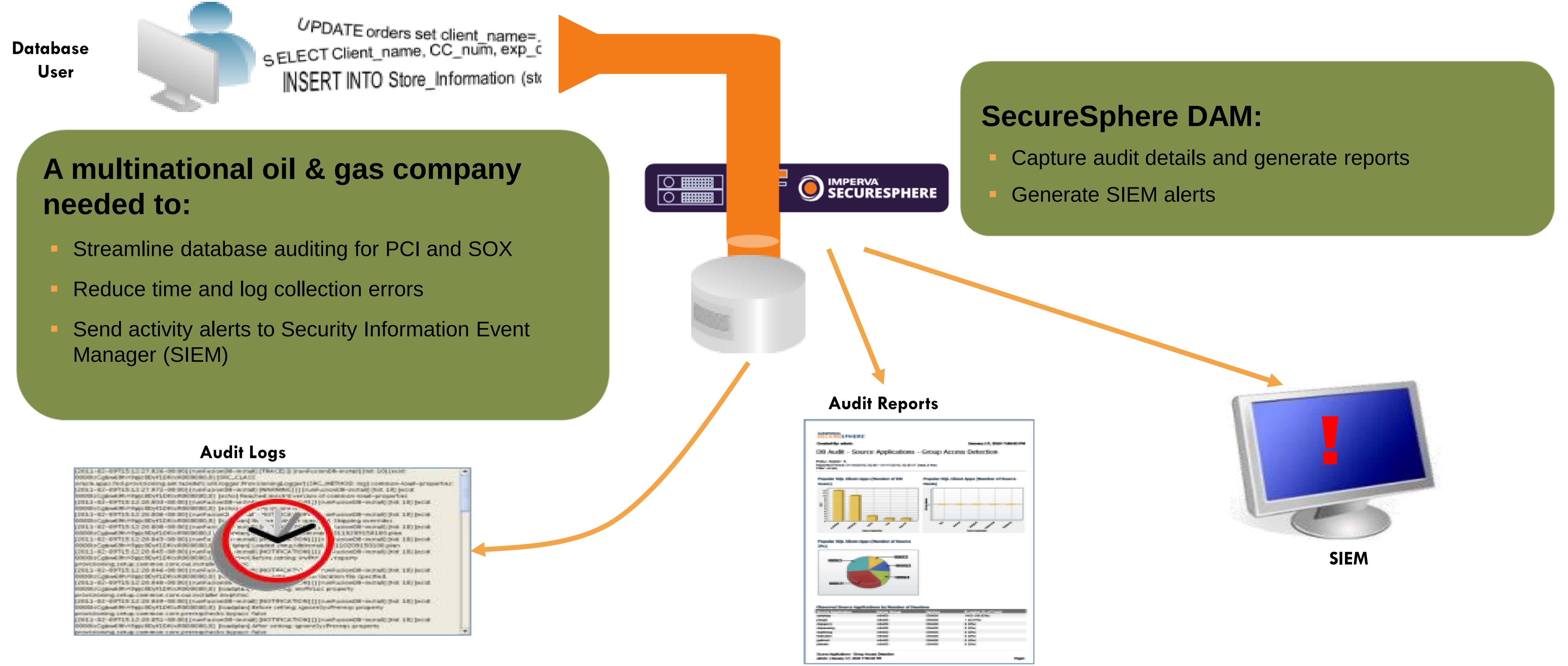
Key setup tool to automate selecting CMK, generating and encrypting CEK, and uploading key metadata to database

Analyze schema and application queries to detect conflicts (build time)

Static schema analysis tool (SSDT only)

Proactive Monitoring

Sensitive Data Auditing



A multinational oil & gas company needed to:

- Streamline database auditing for PCI and SOX
- Reduce time and log collection errors
- Send activity alerts to Security Information Event Manager (SIEM)

SecureSphere DAM:

- Capture audit details and generate reports
- Generate SIEM alerts

Database User
UPDATE orders set client_name=.
SELECT Client_name, CC_num, exp_c
INSERT INTO Store_Information (st
IMPERVA SECURESPHERE
Audit Logs
Audit Reports
SIEM

Auditing Sensitive Data

Reporting

Enterprise class reporting framework

- Analyze threats
- Accelerate compliance

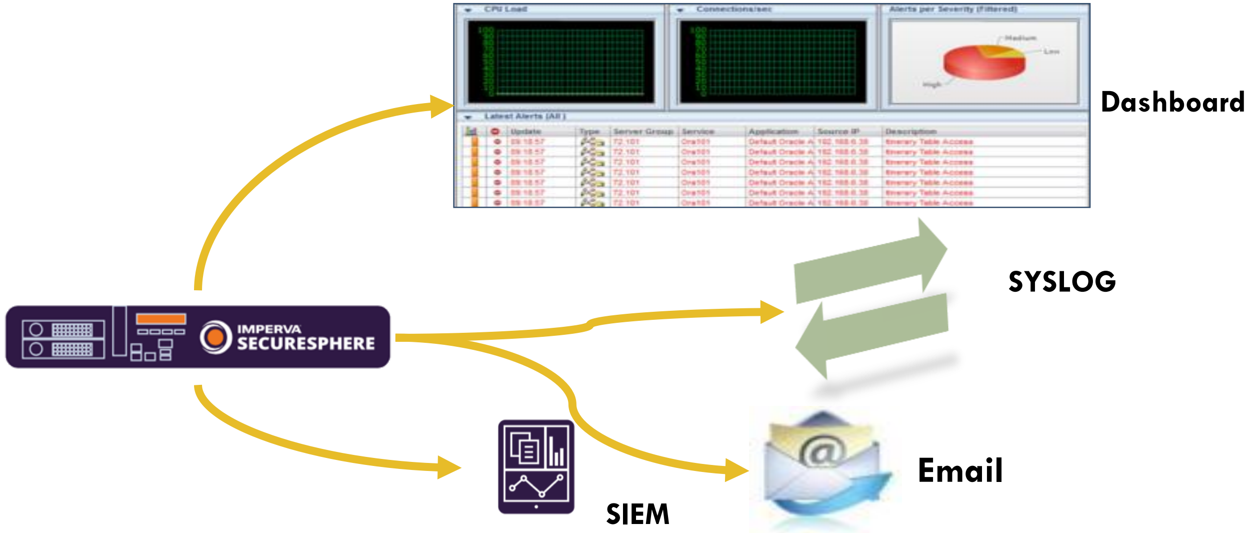
DB Audit - Privileged Query Overview
DB Audit - DDL Commands analysis

PCI, HIPAA, SOX… Custom

Alerting

Alert in real time on suspicious behavior

- Quickly identify attacks
- Prevent data theft



Dashboard

SYSLOG

SIEM

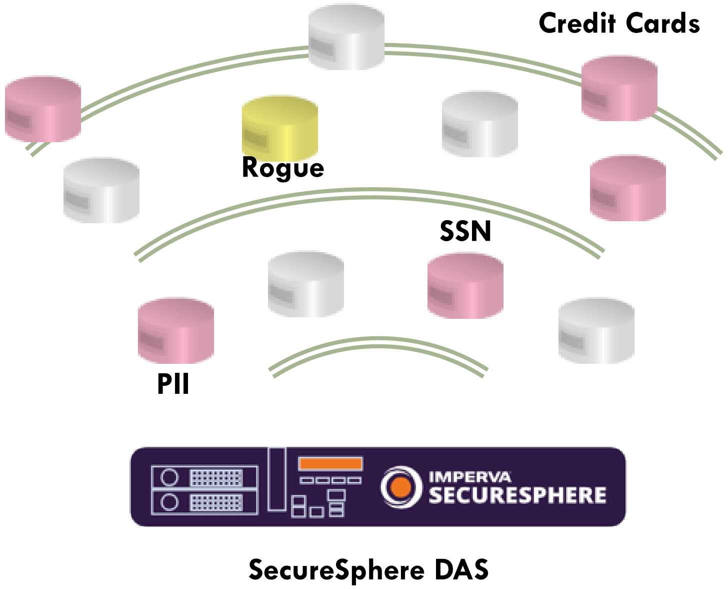
Email

Auditing Sensitive Data

Discovery & Classification

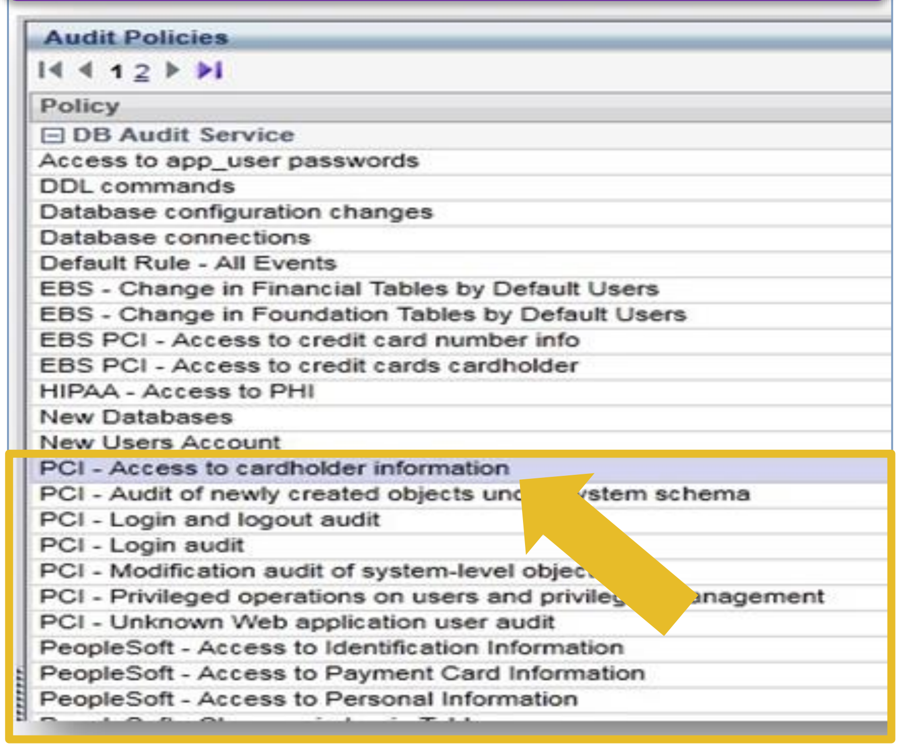
Discover DBs and classify sensitive information

- Discover active DB services
- Identify rogue DBs
- Determine what needs to be monitored



Audit Access to Cardholder Information

Pre-defined PCI Audit Policies



Audit Policies

1 2

Policy

⊟ DB Audit Service

Access to app_user passwords

DDL commands

Database configuration changes

Database connections

Default Rule - All Events

EBS - Change in Financial Tables by Default Users

EBS - Change in Foundation Tables by Default Users

EBS PCI - Access to credit card number info

EBS PCI - Access to credit cards cardholder

HIPAA - Access to PHI

New Databases

New Users Account

PCI - Access to cardholder information

PCI - Audit of newly created objects und ystem schema

PCI - Login and logout audit

PCI - Login audit

PCI - Modification audit of system-level objec

PCI - Privileged operations on users and privileg anagement

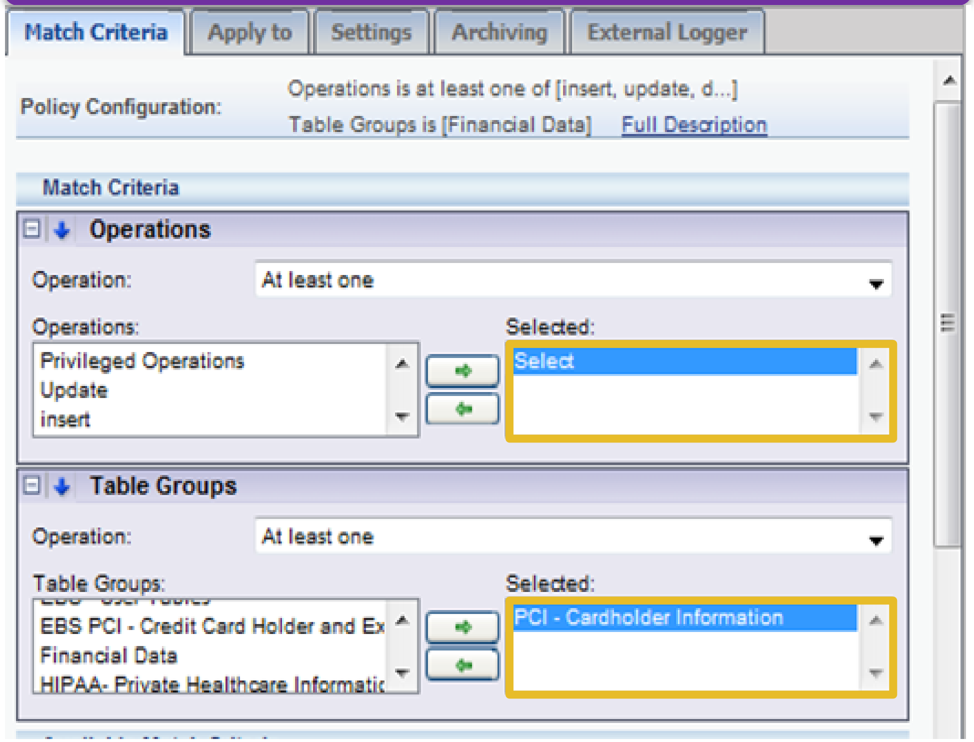
PCI - Unknown Web application user audit

PeopleSoft - Access to Identification Information

PeopleSoft - Access to Payment Card Information

PeopleSoft - Access to Personal Information

SELECT Operations on Cardholder Data



Policy Configuration: Operations is at least one of [insert, update, d...]
Table Groups is [Financial Data] Full Description

Match Criteria

⊟ Operations

Operation: At least one

Operations:
Privileged Operations
Update
insert

Selected:
Select

⊟ Table Groups

Operation: At least one

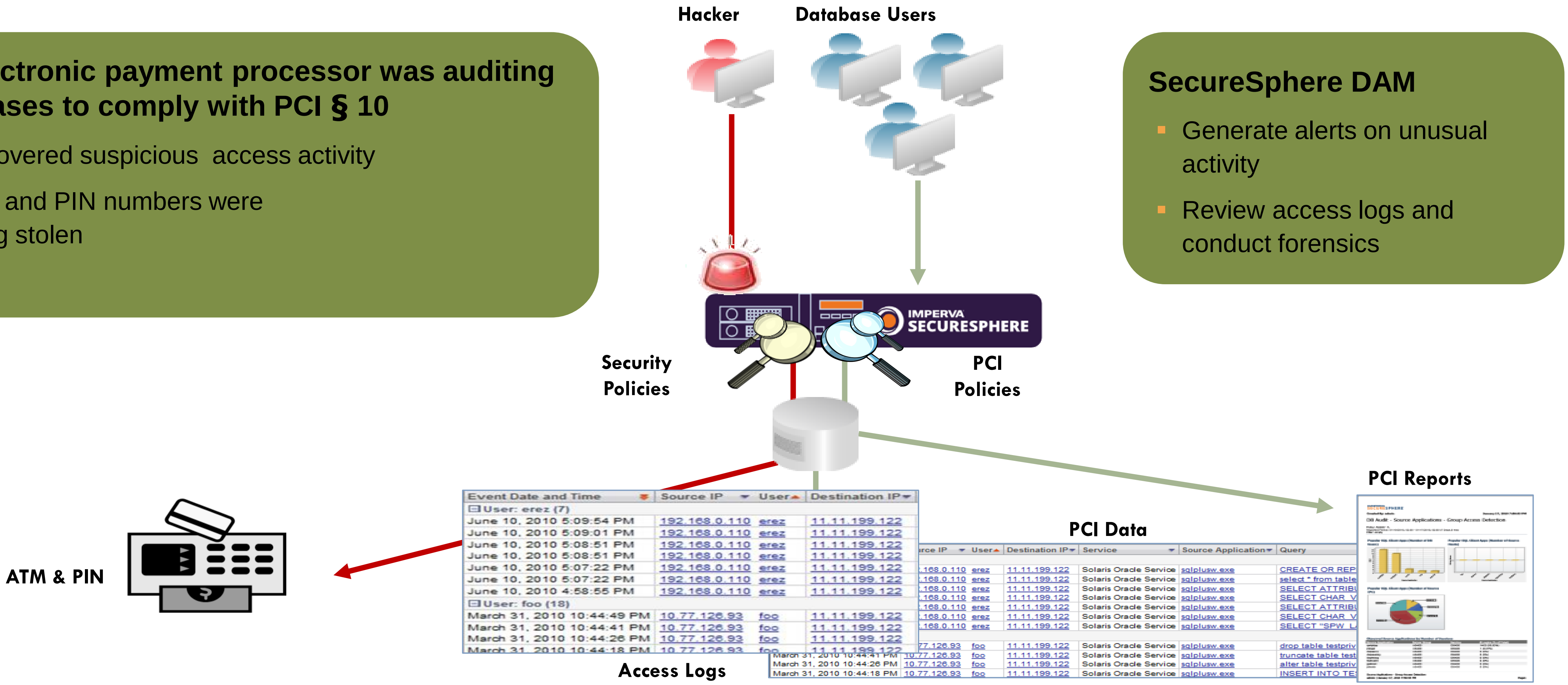
Table Groups:
EBS PCI - Credit Card Holder and Ex
Financial Data
HIPAA- Private Healthcare Informatic

Selected:
PCI - Cardholder Information

Data Theft Prevention

An electronic payment processor was auditing databases to comply with PCI § 10

- Discovered suspicious access activity
- ATM and PIN numbers were being stolen



Access Logs

Event Date and Time	Source IP	User	Destination IP
⊟ User: erez (7)			
June 10, 2010 5:09:54 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 5:09:01 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 5:08:51 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 5:08:51 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 5:07:22 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 5:07:22 PM	192.168.0.110	erez	11.11.199.122
June 10, 2010 4:58:55 PM	192.168.0.110	erez	11.11.199.122
⊟ User: foo (18)			
March 31, 2010 10:44:49 PM	10.77.126.93	foo	11.11.199.122
March 31, 2010 10:44:41 PM	10.77.126.93	foo	11.11.199.122
March 31, 2010 10:44:26 PM	10.77.126.93	foo	11.11.199.122
March 31, 2010 10:44:18 PM	10.77.126.93	foo	11.11.199.122

PCI Data

	rce IP	User	Destination IP	Service	Source Application	Query
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	CREATE OR REP
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	select * from table
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	SELECT ATTRIBU
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	SELECT CHAR V
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	SELECT ATTRIBU
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	SELECT CHAR V
	.168.0.110	erez	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	SELECT "SPW L
	77.126.93	foo	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	drop table testpriv
March 31, 2010 10:44:41 PM	10.77.126.93	foo	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	truncate table test
March 31, 2010 10:44:26 PM	10.77.126.93	foo	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	alter table testpriv
March 31, 2010 10:44:18 PM	10.77.126.93	foo	11.11.199.122	Solaris Oracle Service	sqlplusw.exe	INSERT INTO TE

SecureSphere DAM

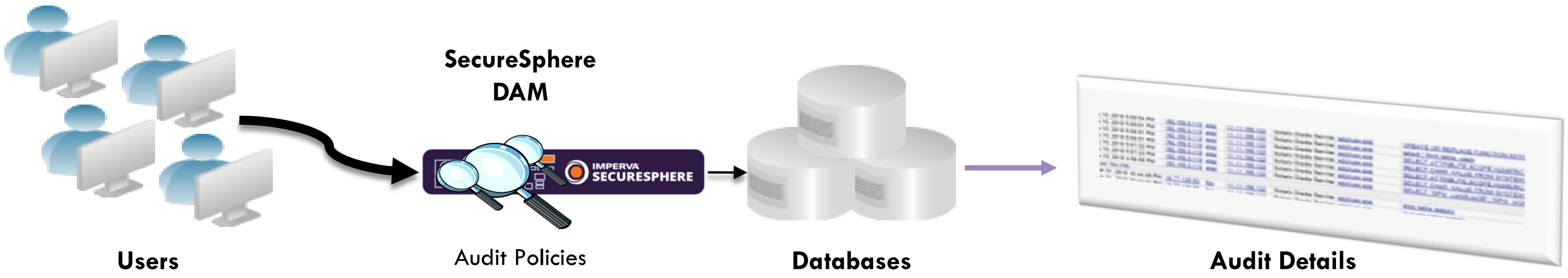
- Generate alerts on unusual activity
- Review access logs and conduct forensics

Data Theft Protection

Activity Auditing

Collect and record database activity details

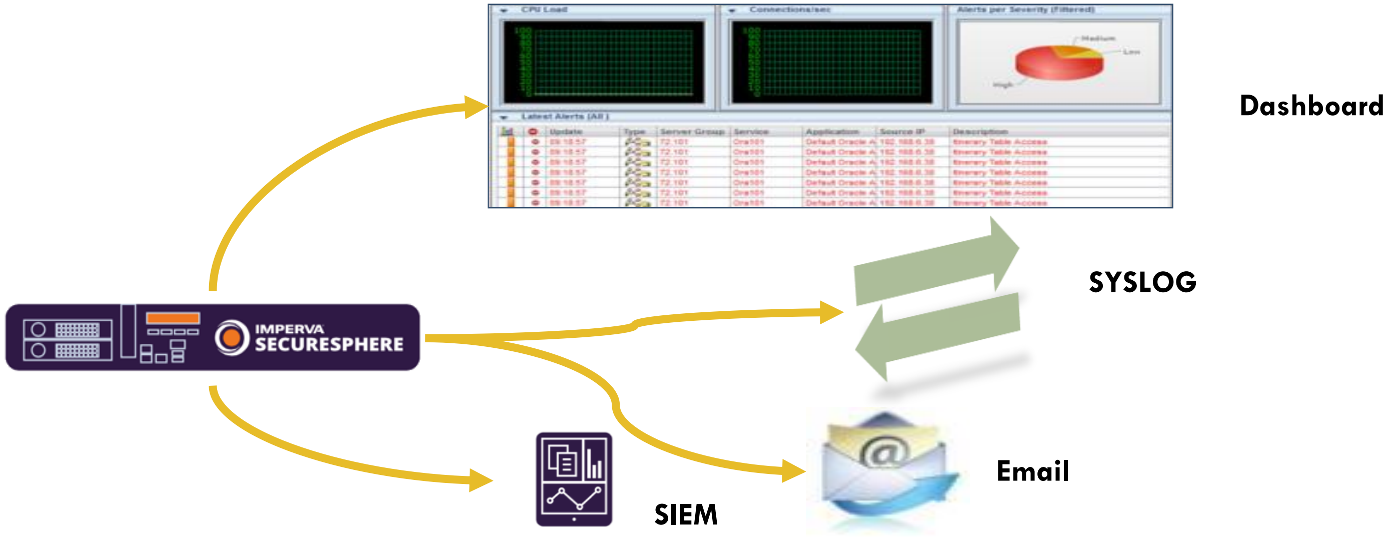
- Satisfy compliance requirements
- Conduct forensic analysis
- Generate alerts



Alerting

Alert in real time on suspicious behavior

- Quickly identify attacks
- Prevent data theft

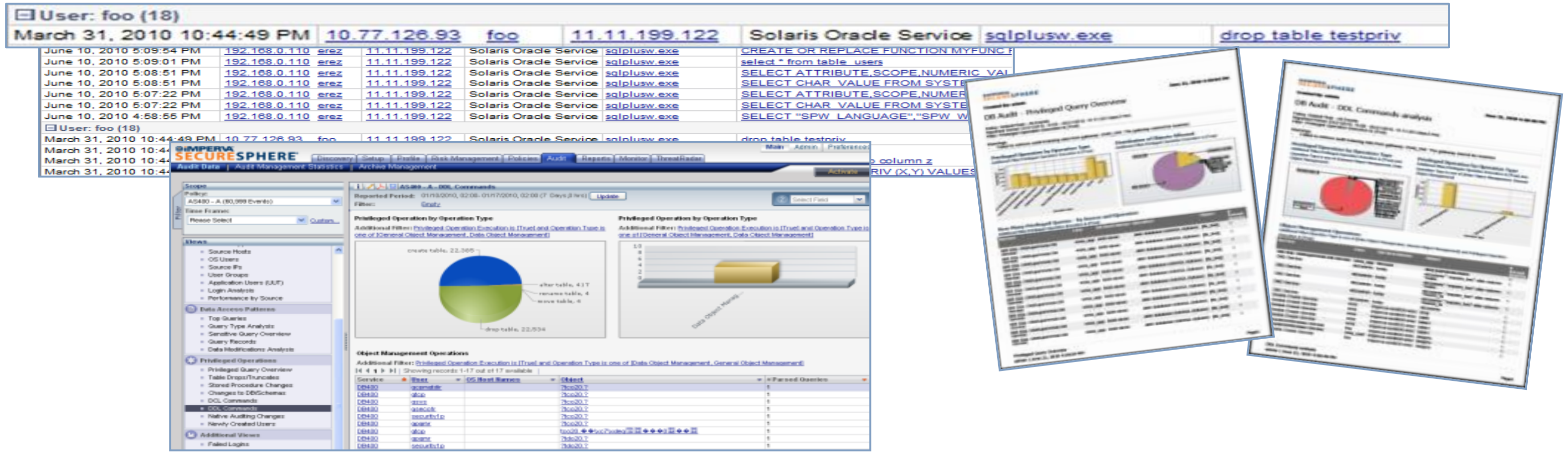


Data Theft Protection

Analytics

Examine detailed audit logs, interactive dashboard views, and reports

- Accelerate forensic analysis
- Simplify compliance



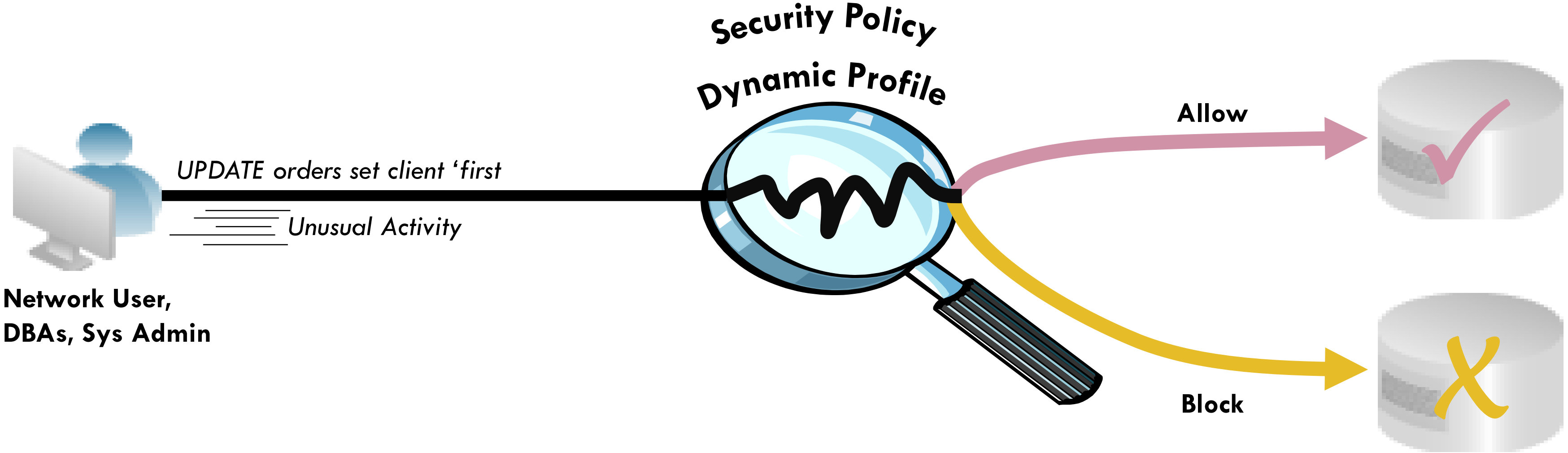
```
User: foo (18)
March 31, 2010 10:44:49 PM   10.77.126.93   foo    11.11.199.122   Solaris Oracle Service   sqlplusw.exe   drop table testpriv
June 10, 2010 5:09:54 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   CREATE OR REPLACE FUNCTION MYFUNC F
June 10, 2010 5:09:01 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   select * from table_users
June 10, 2010 5:08:51 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   SELECT ATTRIBUTE,SCOPE,NUMERIC_VAL
June 10, 2010 5:08:51 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   SELECT CHAR_VALUE FROM SYSTE
June 10, 2010 5:07:22 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   SELECT ATTRIBUTE,SCOPE,NUMER
June 10, 2010 5:07:22 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   SELECT CHAR_VALUE FROM SYSTE
June 10, 2010 4:58:55 PM   192.168.0.110   erez   11.11.199.122   Solaris Oracle Service   sqlplusw.exe   SELECT "SPW_LANGUAGE","SPW_W
User: foo (18)
March 31, 2010 10:44:49 PM   10.77.126.93   foo    11.11.199.122   Solaris Oracle Service   sqlplusw.exe   drop table testpriv
March 31, 2010 10:4
March 31, 2010 10:4                                                                                    o column z
March 31, 2010 10:4                                                                                    RIV (X,Y) VALUES
```

IMPERVA SECURESPHERE
Discovery
Setup
Profile
Risk Management
Policies
Audit
Reports
Monitor
ThreatRadar
Audit Data
Audit Management Statistics
Archive Management
Privileged Operation by Operation Type
Object Management Operations
DB Audit - Privileged Query Overview
DB Audit - DDL Commands analysis

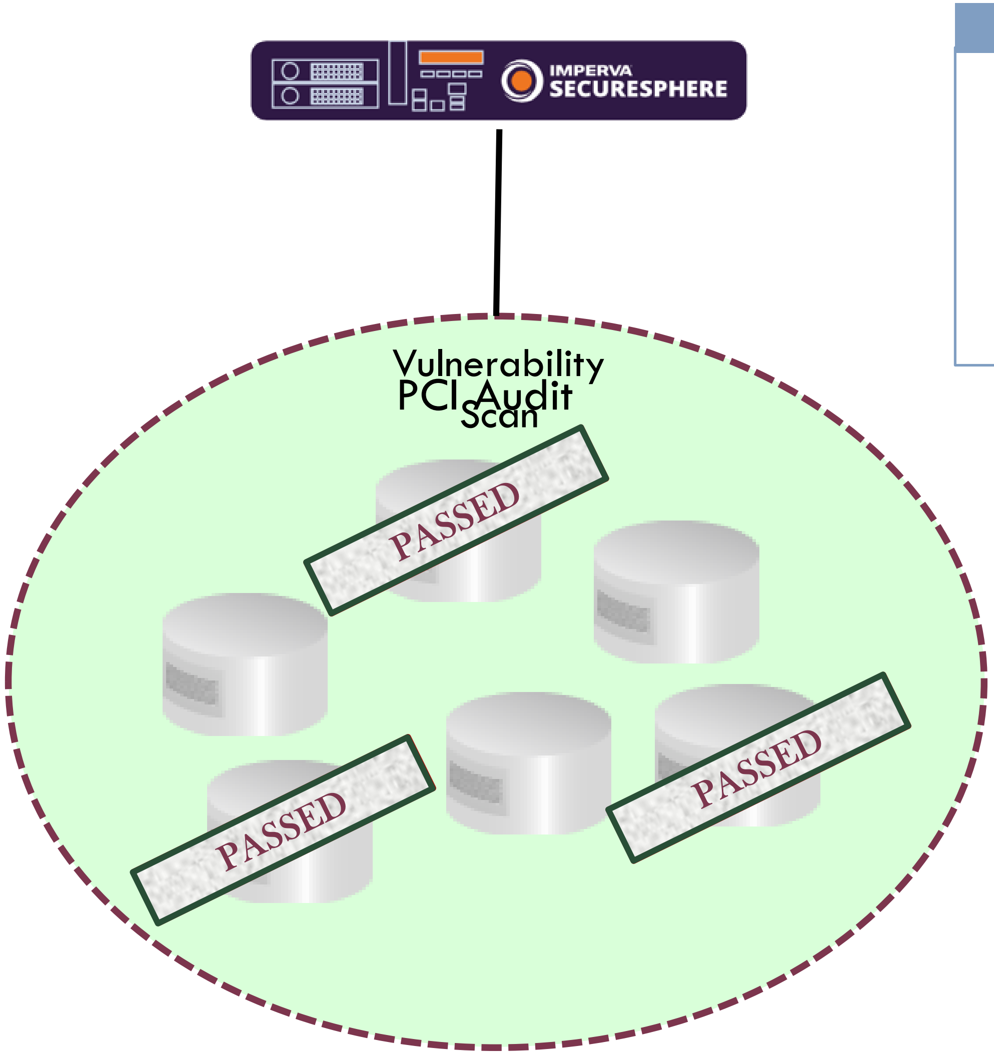
Blocking

Monitor database access

- Prevent unauthorized database access
- Secure sensitive data



Database Vulnerability



An online retailer failed PCI and internal audits:

- PCI 6.1 required quarterly audits
- 300 databases in scope
- Patching was time consuming and disruptive

SecureSphere DAS:

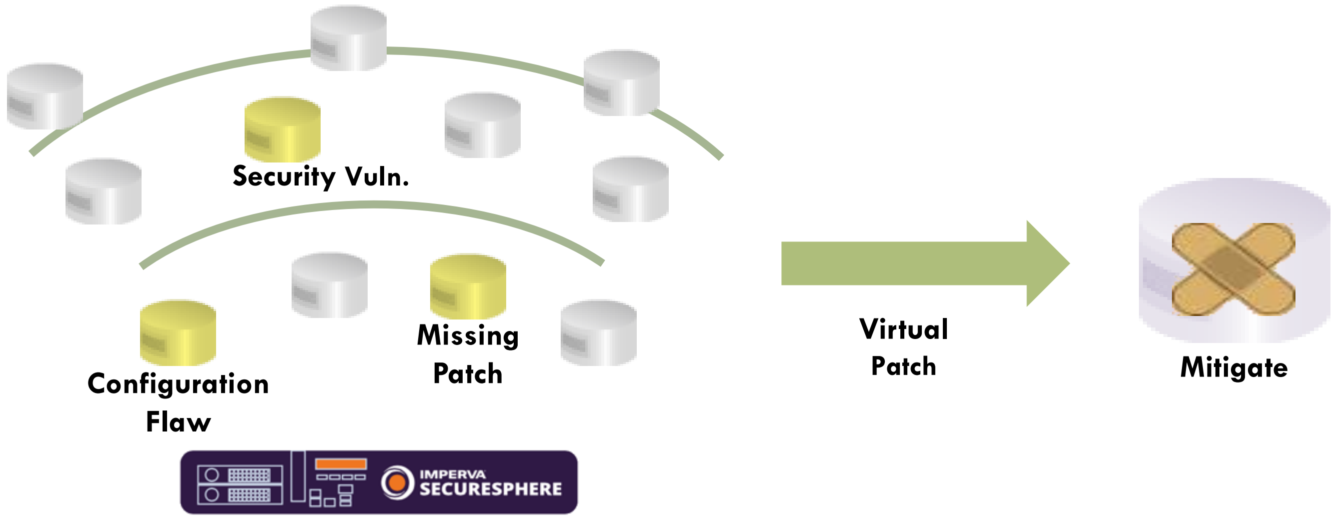
- Database vulnerability scans
- Identify missing patches
- Reduce audit activity to 2 times per year

Database Vulnerability

Vulnerability Scanning & Patching

Identify and mitigate security vulnerabilities and config. flaws

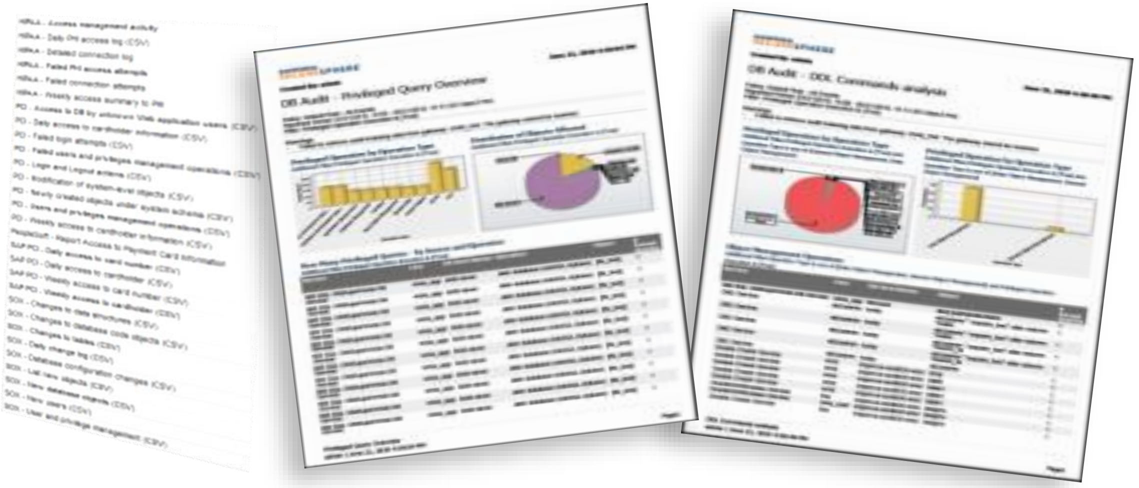
- Automate vulnerability assessment, remediation and verification process



Reporting

Enterprise class reporting framework

- Analyze threats
- Accelerate compliance

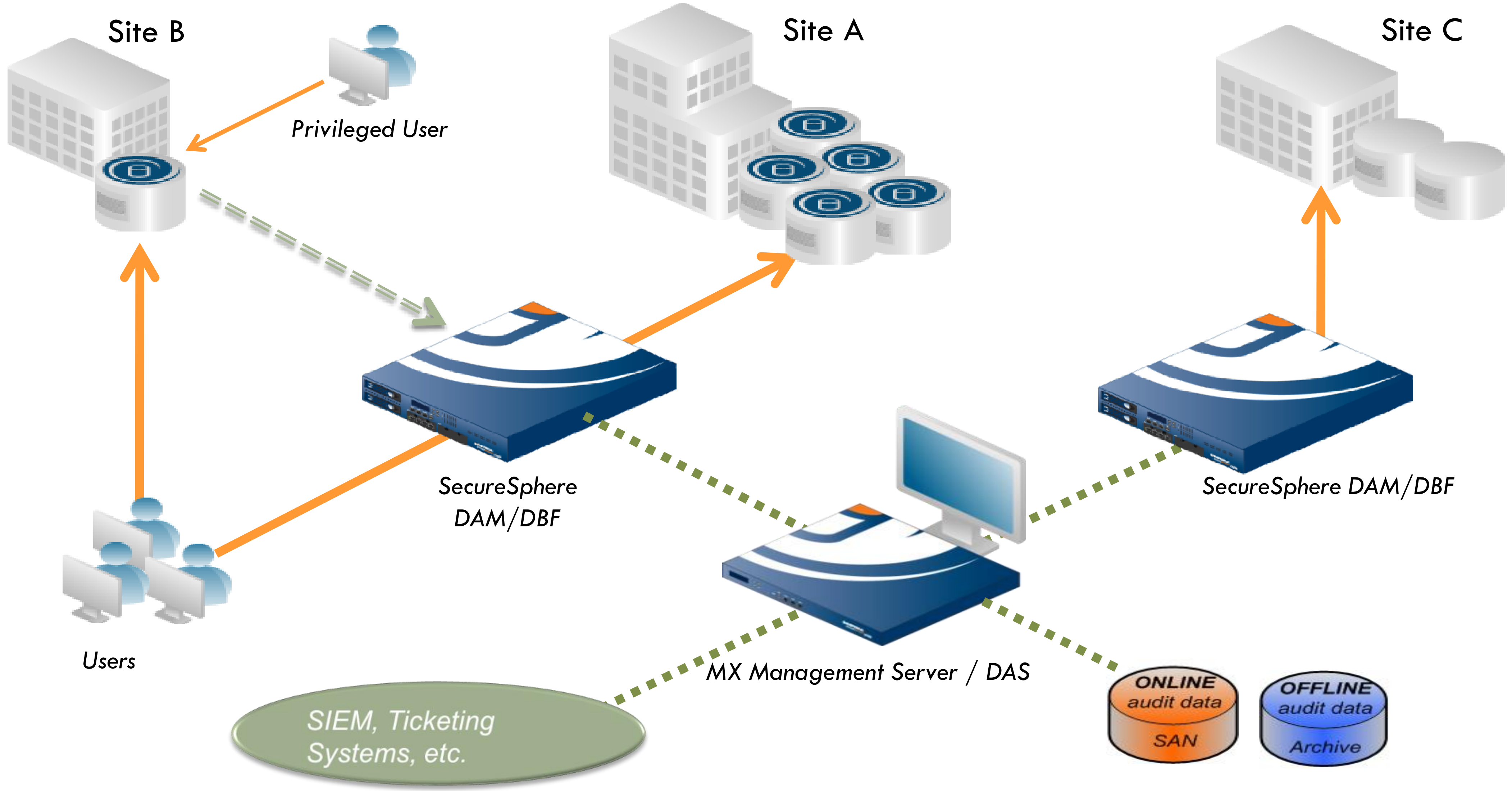

DB Audit - Privileged Query Overview
DB Audit - DDL Commands analysis

PCI, HIPAA, SOX…

Custom

Content

- What is Database Security?
- What is Database Security Technical?
- **How to deploy DBF?**


Site B
Privileged User
Site A
Site C
SecureSphere DAM/DBF
SecureSphere DAM/DBF
Users
MX Management Server / DAS
SIEM, Ticketing Systems, etc.
ONLINE audit data
SAN
OFFLINE audit data
Archive

Question ???