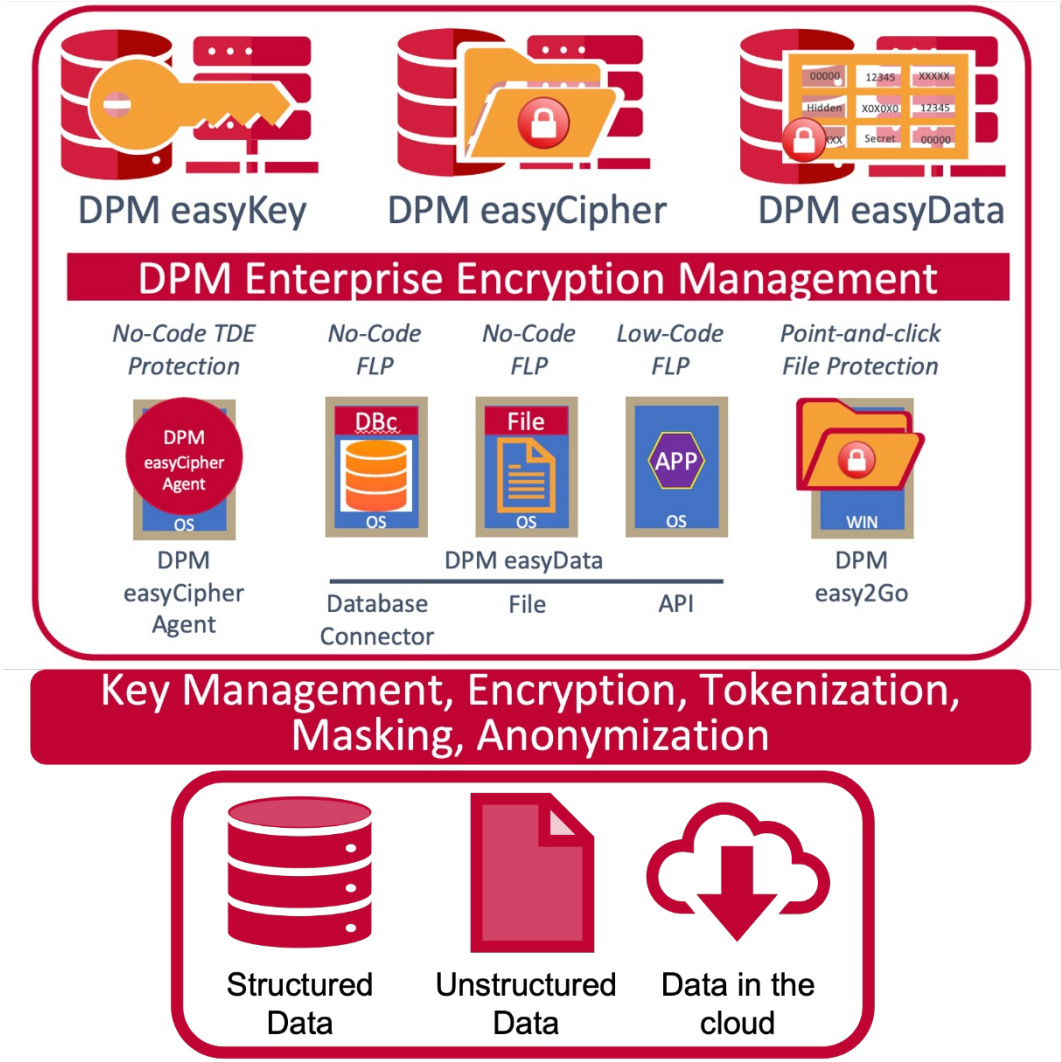
Date: August, 2023

Randtronics Data Privacy Manager

Masking and Tokenization Use Cases

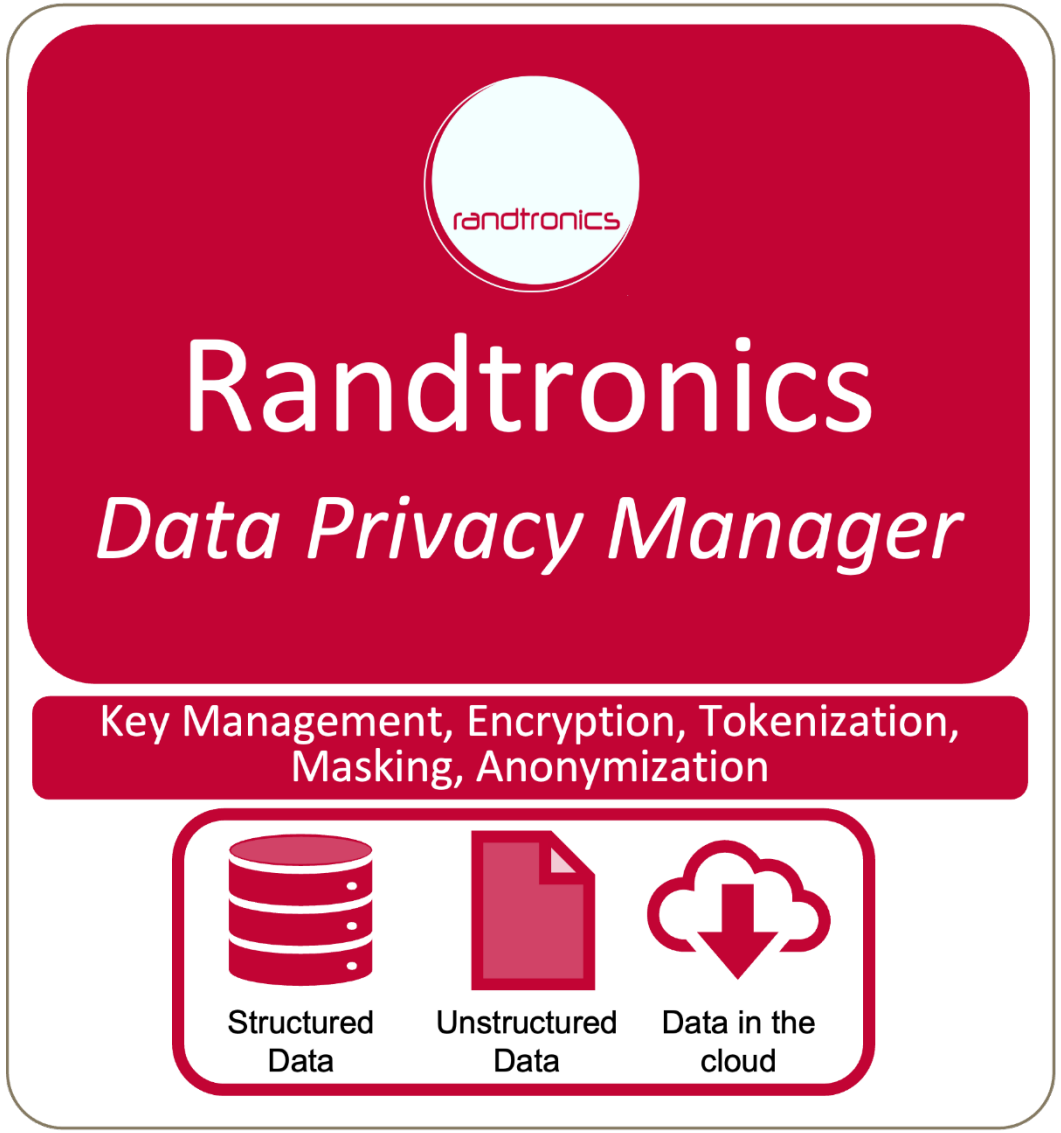


Introducing: Randtronics DPM

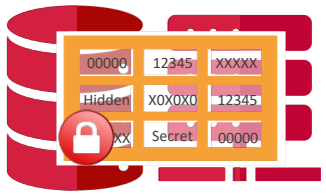
- DPM unifies and simplifies encryption across file systems, databases and applications
- Standardized Encryption protection
  - Uniform protection for all systems and environments
  - Encryption complexity handled by DPM
- Data privacy team empowered to define and implement encryption policies for the whole organization

Key Benefits

- Flexible range of masking and tokenization methods supporting wide-range of use-cases
- API level methods can be easily integrated into any system with minor code changes and allows data to be safely stored on any back-end system
- No-code options available for Oracle and MS-SQL databases and flat-file redaction



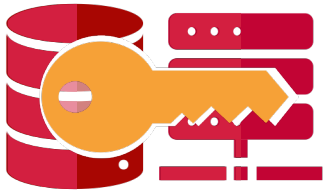
Randtronics DPM Masking & Tokenization with enterprise key management



DPM easyData

- Field level data de-identification
- Masking, Tokenization and Encryption
- API for code-level data protection
- Database Connector for Oracle & MS-SQL Server databases
- Flat file tokenizer

Dependency – easyData requires easyKey to also be installed



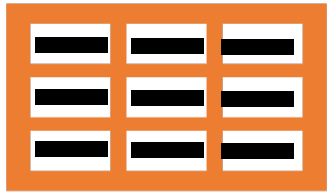
DPM easyKey

- Enterprise Key and Certificate Manager
- 100% software with optional multi-vendor HSM integration

Encryption and Data De-identification: Basic concepts

Encryption

Document



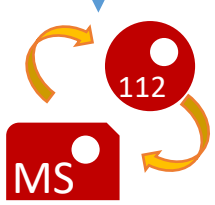
Database

Data De-identification

Mary.Stewart1542@gmail.com

#######@gmail.com

Masking



Tokenization

Somebody.else@gmail.com

MaryStewart1542@gmail.com

Dictionary Tokenization

Reversibility

Strict Non-Reversible

Non- Reversible

Reversible

Encryption – offers the strongest form of protection for column/field level data however:

- Format preserving encryption option enables data protection without creating problems with strongly-typed DB fields, but
- it is pretty hard to operate if all data remotely sensitive is 'blacked out'

Data De-identification - non-encryption methods for protecting content in databases and files:

- Masking – pattern partially or fully replaces data
- Dictionary Tokenization – real data replaced with something safe, that looks similar but has been pulled from a pre-defined list
- Tokenization – voucher/ 'cloakroom ticket', substitute data with a token. Token can be alpha; numeric or random

Reversibility – capacity to recover original.

- Reversible – encrypted data can be decrypted, tokenized data can be restored from token vault
- Non-reversible – original data not available as part of normal operations. If required data-administer can manually retrieve original data from central token-vault
- Strict Non-reversible – encryption key is destroyed, or token pair is destroyed in token vault – original data can not be recovered

randtronics

Data protection: example

Name: Mary Stewart
Gender: Female
eMail: Mary.Stewart1542@gmail.com
Bank Account Number : 9578 2318
Credit Card: 4487 8239 271

Name: Robert Scott
Gender: ###
eMail: ####.#####1542@gmail.com
Bank Account Number : #### 2318
Credit Card: 9987 3621 349

Dictionary Tokenization :

- Picked from a dictionary of names or other list of fake values
- In this example we have de-identified gender
- Remains easy for human staff to work with
- Complies with name field validation rules

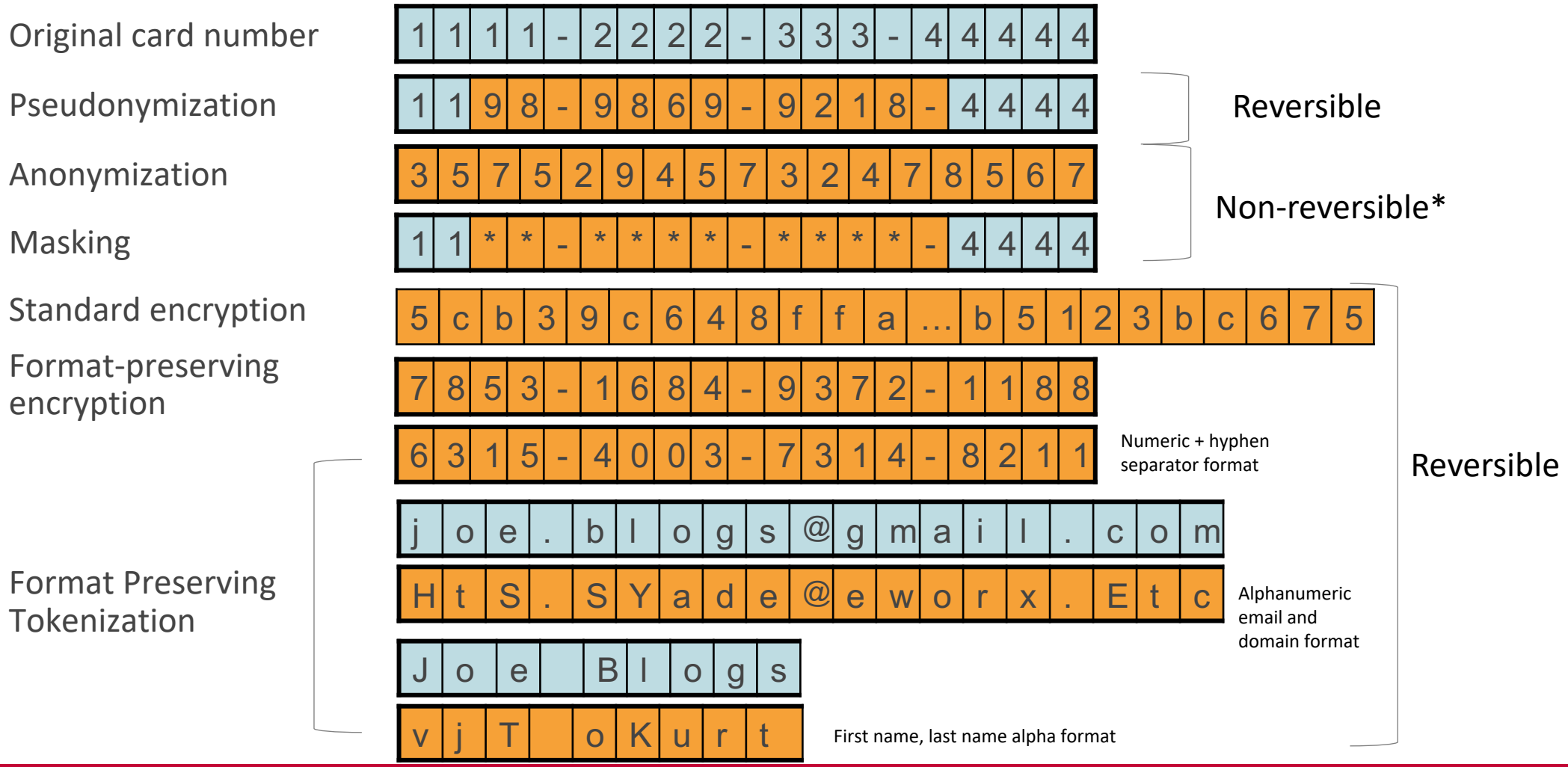
Masked - gender, email, bank account number:

- Data obscured in an obvious way
- Mask selected to protect information whilst enabling staff to perform role
  - Last 4 digits of bank account
  - Distinguish between email addresses
  - Recognize domain address

Tokenized Credit Card

- Substituted number that satisfies data type rules
- Original data stored with token in token vault

Data De-identification Techniques: Examples



*Non-reversible = original data can not be directly recovered in normal operations.
DPM easyData offers options of non-reversible data protection via a) consistent token or b) single-use token.
Consistent token, by its very nature, requires token pair to be stored in central token vault

Format preserving Tokenization options

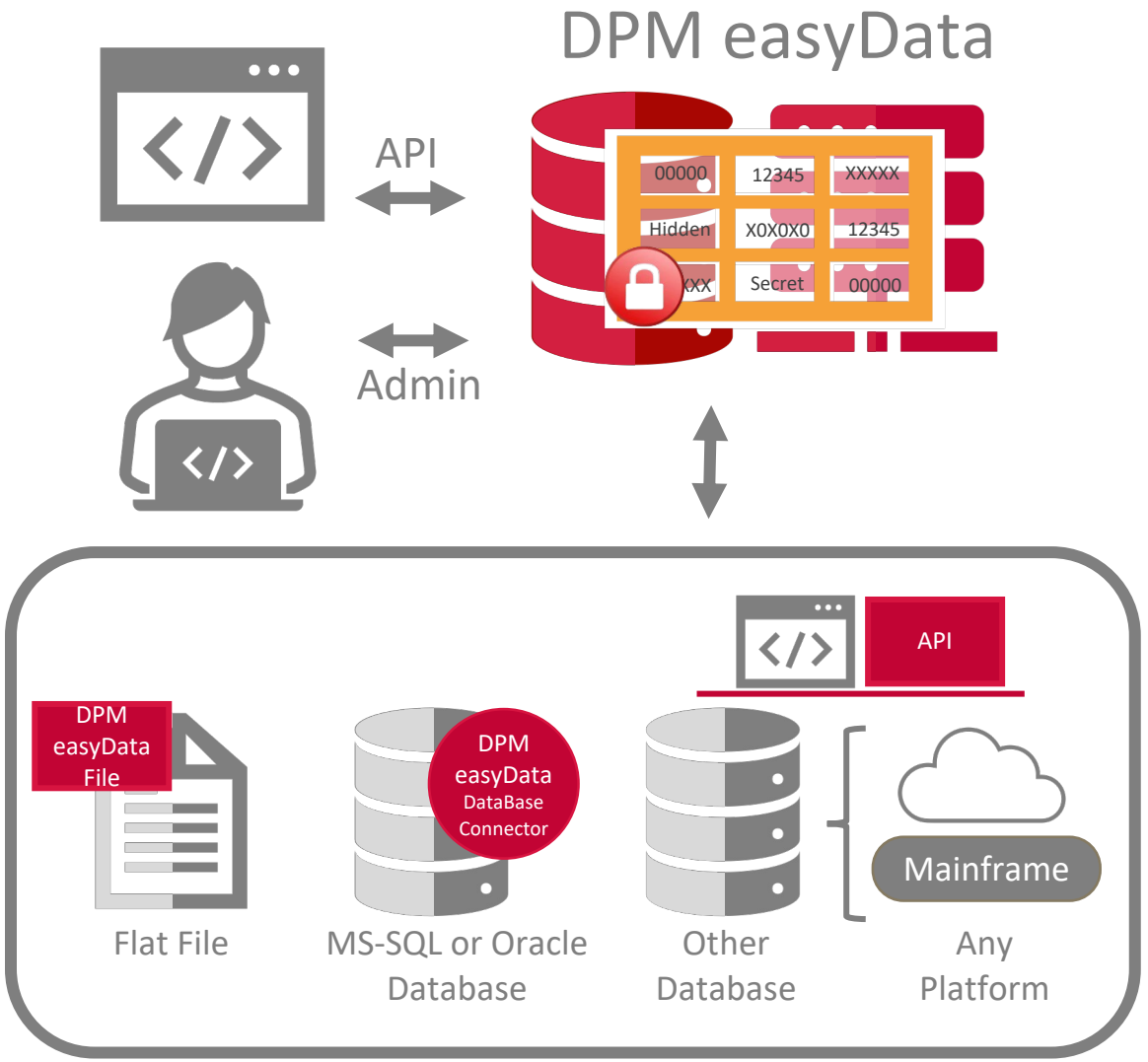
Original card number	1 1 1 1 2 2 2 2 3 3 3 4 4 4 4 4
Numeric	6 3 1 5 4 0 0 3 7 3 1 4 8 2 1 1
Preserving some characters	1 1 9 8 9 8 6 9 9 2 1 8 4 4 4 4
Alphanumeric	1 1 A B 9 8 c 9 F 2 1 8 4 4 4 4
With delimiters	1 1 - 8 9 8 6 9 9 2 1 - 4 4 4 4
Dictionary (predefined list)	4 6 5 3 0 4 1 8 6 7 2 4 1 6 7 4
Language - German	1 1 ä A P ß k E C ü t s 4 4 4 4
Language - Japanese	あ 日 青 か 菊 笹 身 べ 背 ア 郎

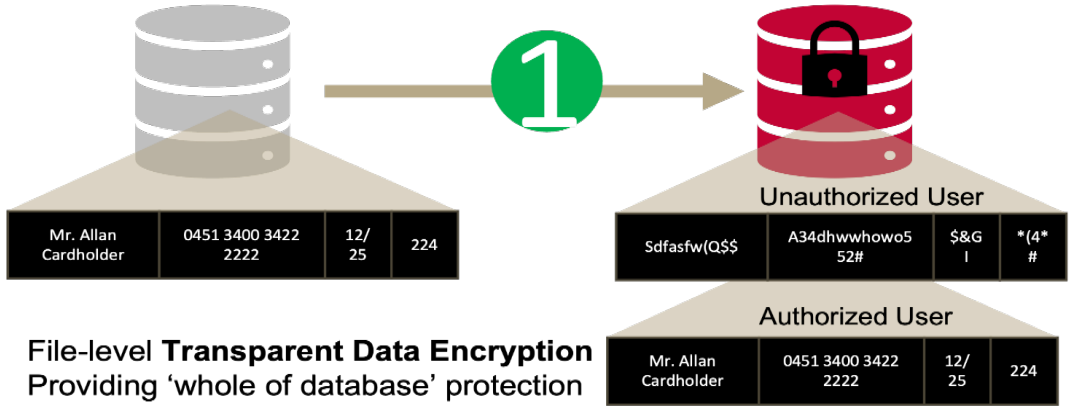
randtronics

DPM easyData overview

Centralized Masking & Tokenization

- Protects field-level data via encryption, tokenization or masking
- Accessed via API, or via connectors (MS-SQL/Oracle database or Flat File)
- High performance, scalable and rapidly deployable
- Centralized Management & Control of data protection policy and key management
- Granular Protection levels to Root, Privileged & Authorized users

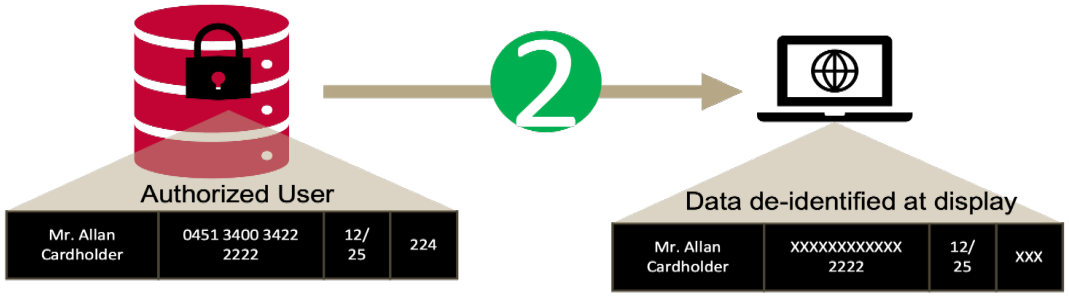




File-level **Transparent Data Encryption**
Providing 'whole of database' protection

DPM easyCipher

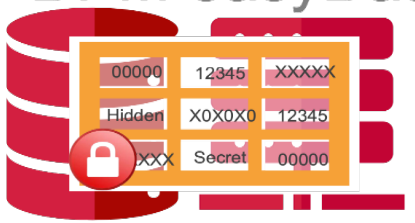
TDE



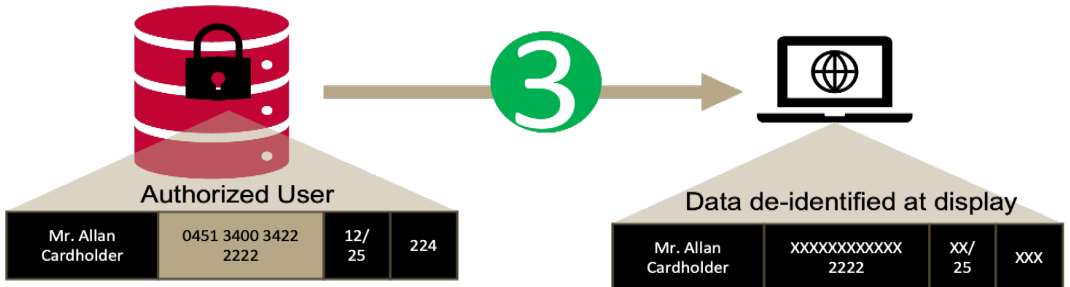
Field-level Masking
Protecting PAN data when displayed

PCI DSS Compliance: *Good, Better, Best*

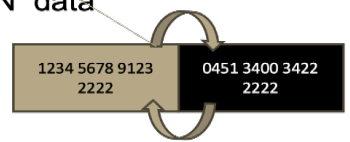
DPM easyData



Tokenization



Tokenization Masking
Minimize storage of PAN data



Use case #1: PCI DSS Compliance

Problem

- Organizations storing credit card data need to protect card holder information

Solution

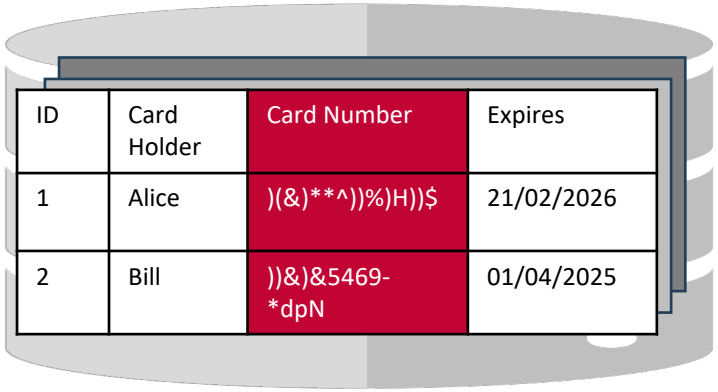
- Create policy within DPM easyData to encrypt PAN details as they are written to database
- Create policy within DPM easyData to mask PAN details when displayed

Clear data

ID	Card Holder	Card Number	Expires
1	Alice	5567 8911 5657	21/02/2026
2	Bill	9371 5297 9008	01/04/2025

Data stored in Database

Encrypted data



ID	Card Holder	Card Number	Expires
1	Alice	)(&)**^))%)H))$	21/02/2026
2	Bill	))&)&5469-*dpN	01/04/2025

Masked data

ID	Card Holder	Card Number	Expires
1	Alice	#### #### 5657	21/02/2026
2	Bill	#### #### 9008	01/04/2025

Use case #2: Reduce PCI DSS Compliance Scope

Problem

- Organizations storing credit card data need to protect card holder information
- Credit card data stored on multiple systems, even if all of these systems use encryption, they still all fall within scope of the 12 Requirements and annual PCI DSS review process
- Organizational exposure to credit card data breach is magnified

Solution

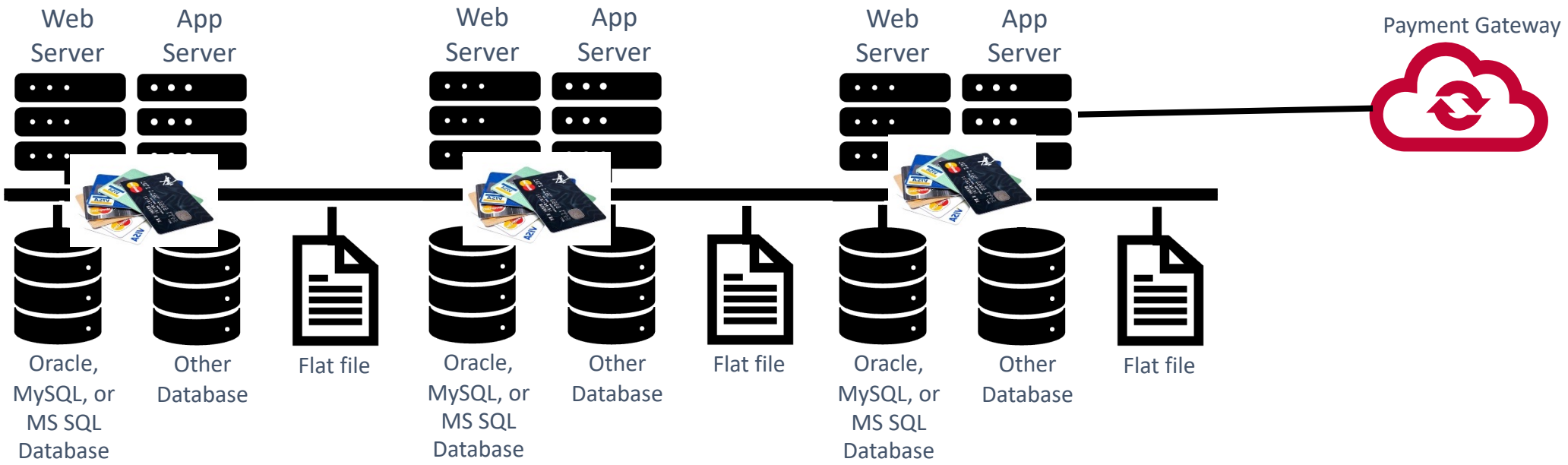
- Create DPM easyData policy to tokenize credit card details (reversible data protection)
- Use tokens in place of credit card details where possible to remove systems from the scope of PCI DSS review process
- Now scope is reduced systems (7 to 1 in this example) thereby reducing initial capital expenditure and recurring financial burden and drain on remediation resources
- Easily demonstrate PCI DSS compliance for these systems

Multiple Systems Storing Credit Card data

Few systems storing Credit Card data

Use case #2: Reduce PCI DSS Compliance Scope – cont.

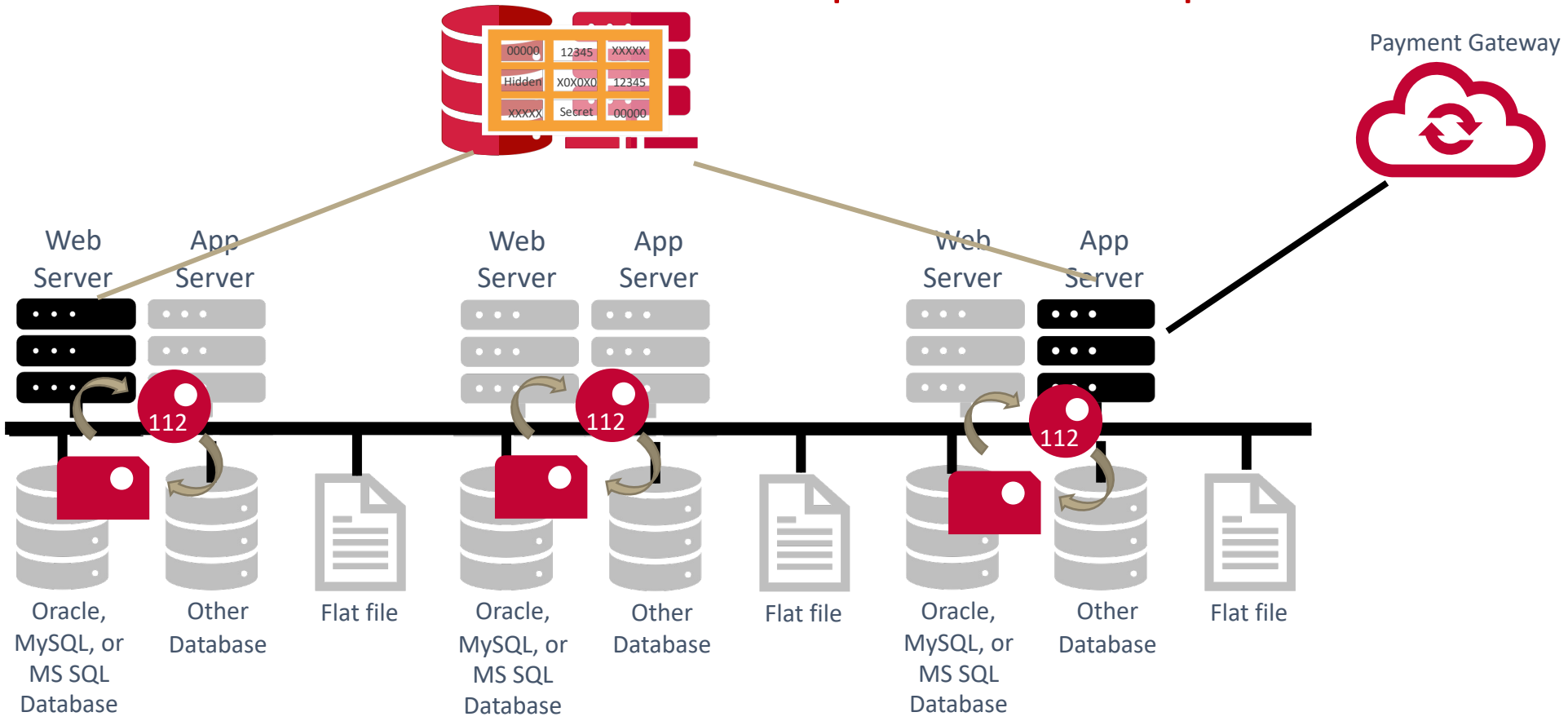
Before



In this example, we have 15 systems handling PAN (storing, processing, transmitting)
All 15 systems are in scope for PCI DSS compliance, that is, all 12 requirements needs to be satisfied
If costs of compliance per system is $10K then recurring fee is $150K

Use case #2: Reduce PCI DSS Compliance Scope – cont.

After



- The card number is de-identified when it is received by the web application
- The web application stores the token, not the card number in the shared database
- Only applications needing the card number are allowed to detokenize it
- 15 systems had access to the card number before, now only 2 systems are in scope
- Costs of compliance before was $150K recurring, now it is $20K

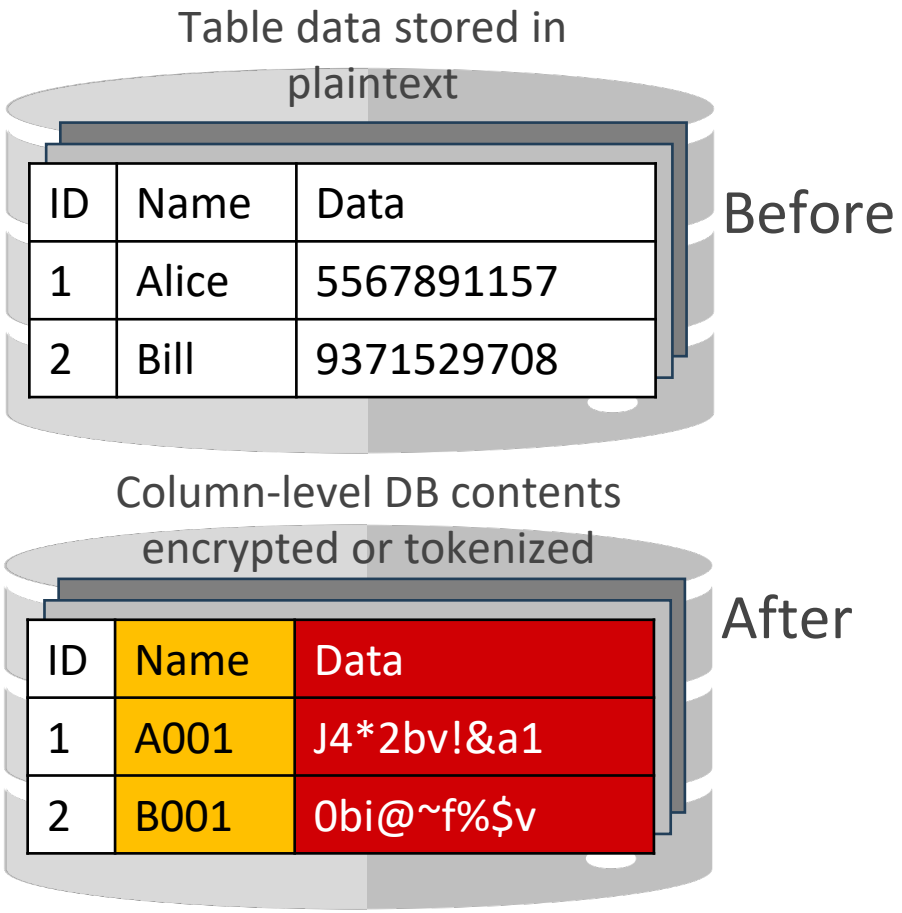
Use case #3: Zero-trust Database content protection

Problem

- Organizations following best-practice ‘zero-trust’ principles, desire to protect sensitive data in databases unauthorized access including from privileged IT staff, system admins, software developers and DBA’s.
- The challenge - how to protect data from DBA whilst preserving data indexes to minimizing impact on system performance.

Solution

- Policy within DPM easyData restricts access to data by encryption or tokenization.
- Policy is applied to data being stored in database by a) call easyData API from application or b) using easyData DB Connector for MS-SQL Server or Oracle DB’s
- Database indexes protected data and DB searches are conducted using tokens – which the application retrieves easyData prior to calling the database.
- Outcome: Application data in databases is protected with minimal performance impact



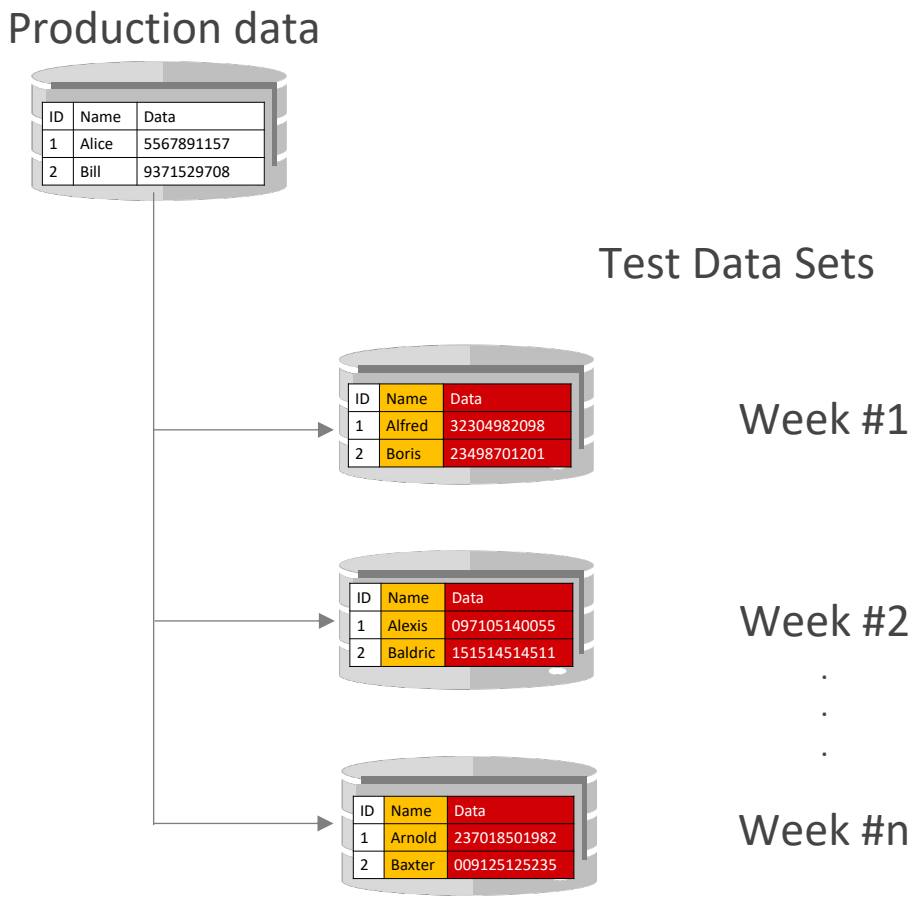
Use case #4: Creating ‘safe’ test data sets

Problem

- Developers need realistic and appropriately large data sets to test against
- Past practice was to take a copy of production data, creating an extra data breach vulnerability
- Continuous need for new test data sets

Solution

- Policy within DPM easyData used to replace clear data with format consistent replacements (non-reversible protection)
- Multiple test databases produced at demand
- Outcome: realistic databases available on-demand without creating a data breach risks



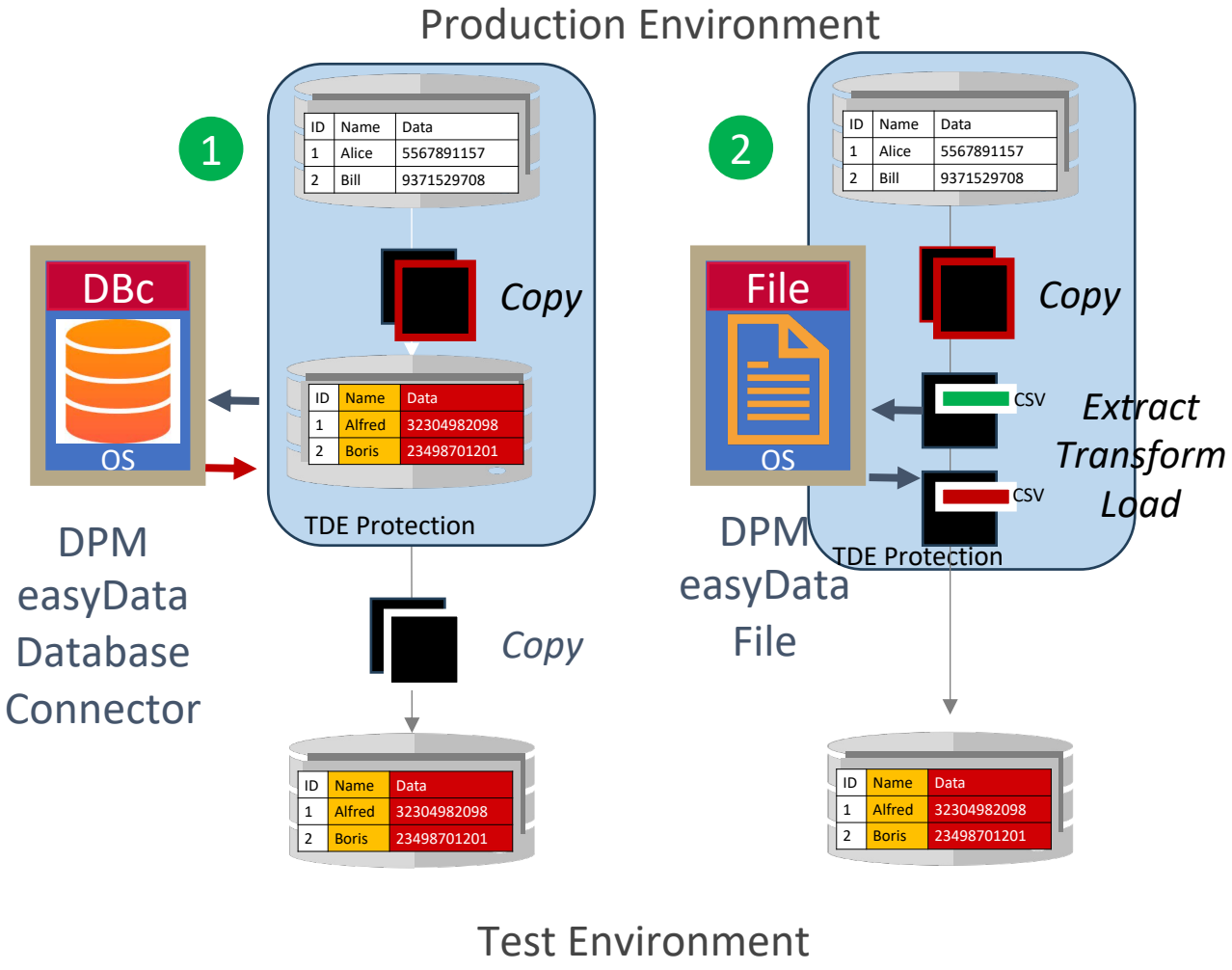
Use case #4: Continued: Protecting Clear Data

Naturally we assume all customers are protecting production databases with TDE.

Using DPM easyData, customers have two methods of creating a sanitized copy of their Production database in their Test environments

Method 1: Use easyData Database Connector and transform sensitive column contents into anonymized values

Method 2: Perform an Extract, Transform, Load (ETL) operation using DPM easyData File to anonymize values in a CSV file



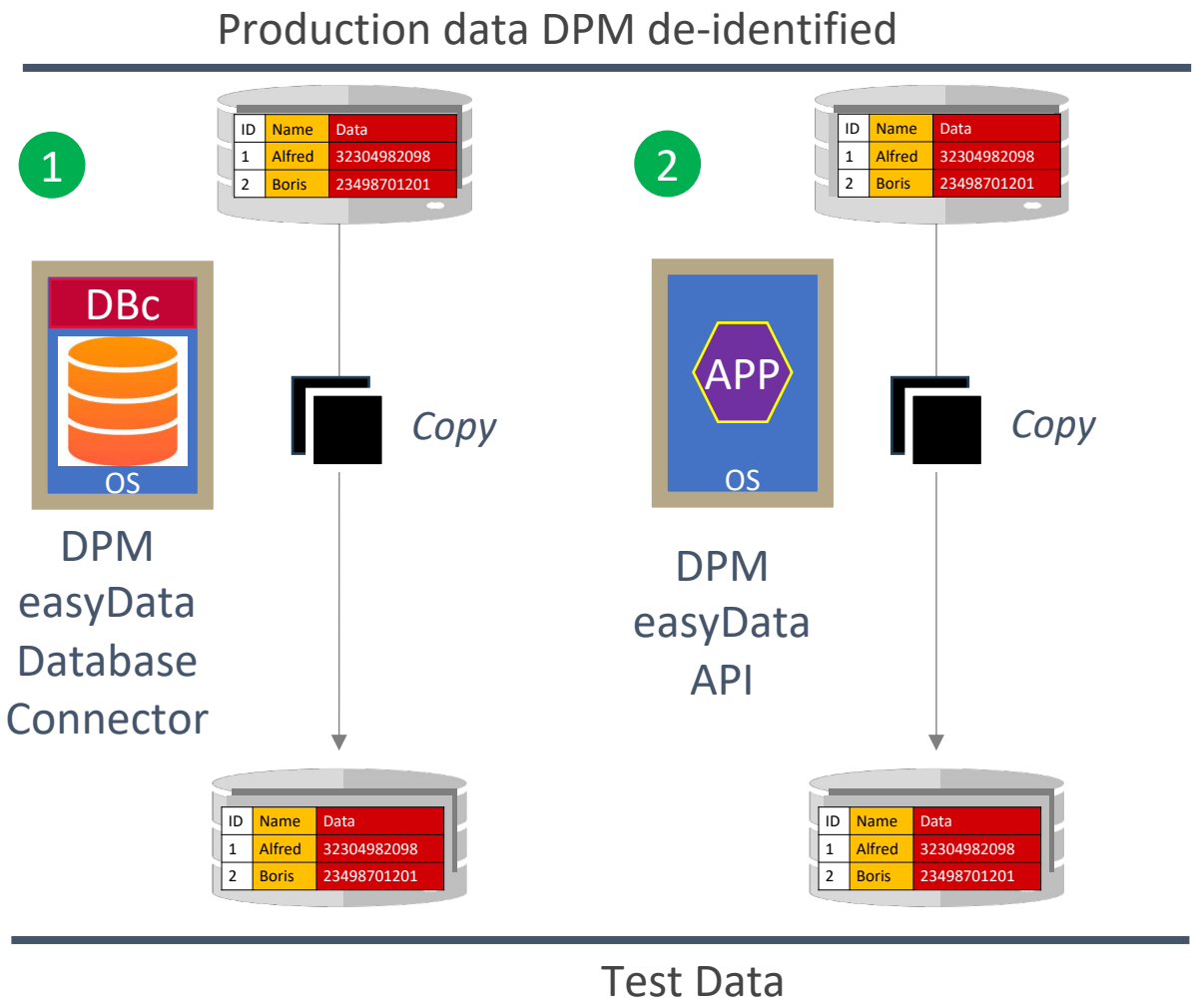
Use case #4: Bonus feature for existing customers

For existing users of DPM easyData, sensitive data columns in production database may already have been replaced with encrypted or tokenized contents:

Sensitive data has has been replaced, either:

- within database via DPM easyData Database connector, or
- external to database via calls to DPM easyData API

Since the contents are already protected, the database can be safely copied into the Test environment



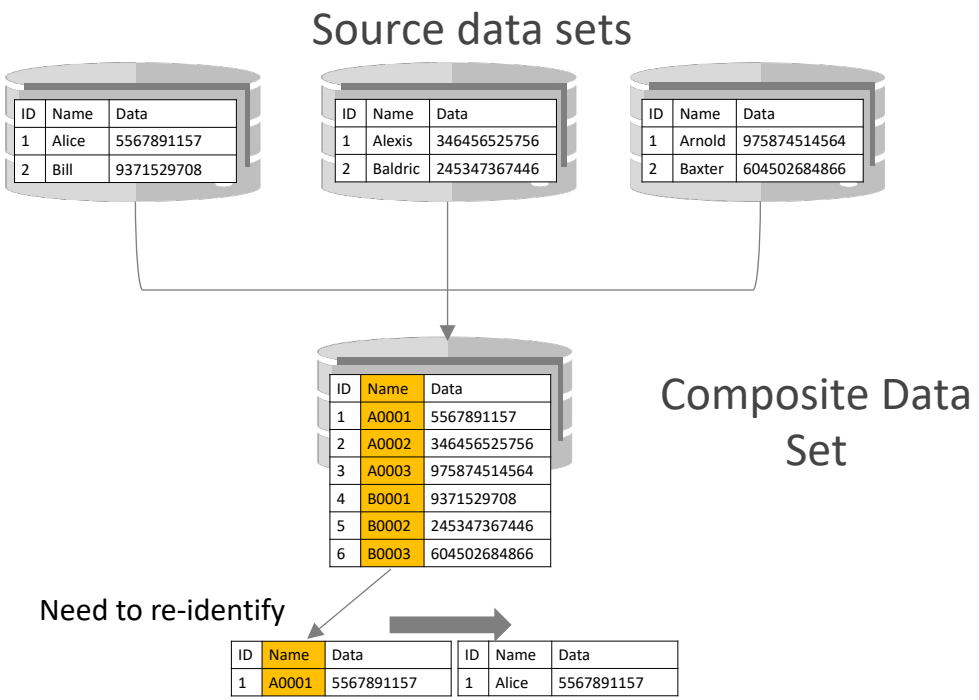
Use case #5: Criminal Justice Analytics

Problem

- Criminal justice organizations wish to combine multiple datasets to identify potential patterns
- In normal use, data needs to be de-identified
- Occasionally, it can be important to re-identify a source record

Solution

- Policy within DPM easyData used to replace clear data from source databases with format consistent replacements (reversible protection)
- Data policy defines if and for how long original data can be retrieved from protected data
- Outcome: data in composite data set is protected for general analytics purposes. Ability to re-identify is available but highly restricted



Use case #6: Witness Protection

Problem

- Law enforcement organizations need to enforce strict controls over identity of some witnesses including obscuring details within operational systems

Solution

- Witness identification details in operational records can be replaced with fake values using Dictionary Tokenization.
- Token Type Option 1: Constant Token. Data replaced across multiple systems using same fake data
- Token Type Option 2: Single use Token. Data replaced across multiple systems using multiple unique versions of fake data.
- Reversibility Options (applies to both token types)
- - reversible – data policy allows original data can be restored by whitelist applications via API call
- - non-reversible - data policy blocks recovery of original data (i.e. intervention of data admin required to obtain original data)

Clear data

ID	Witness Name	Safe House Location	Case Number
1	Alice	5 Suburban Ave, Chicago	123455
2	Bill	12 Midtown, New York	661234

Before

Operational Systems

1

ID	Witness Name	Safe House Location	Case Number
1	Mary	61 Anystreet Ave, Townville	123455
2	Rodger	37 MadeUp Street, Anytown	661234

After

2

ID	Witness Name	Safe House Location	Case Number
1	Mary	61 Anystreet Ave, Townville	123455
2	Rodger	37 MadeUp Street, Anytown	661234

3

ID	Witness Name	Safe House Location	Case Number
1	Mary	61 Anystreet Ave, Townville	123455
2	Rodger	37 MadeUp Street, Anytown	661234

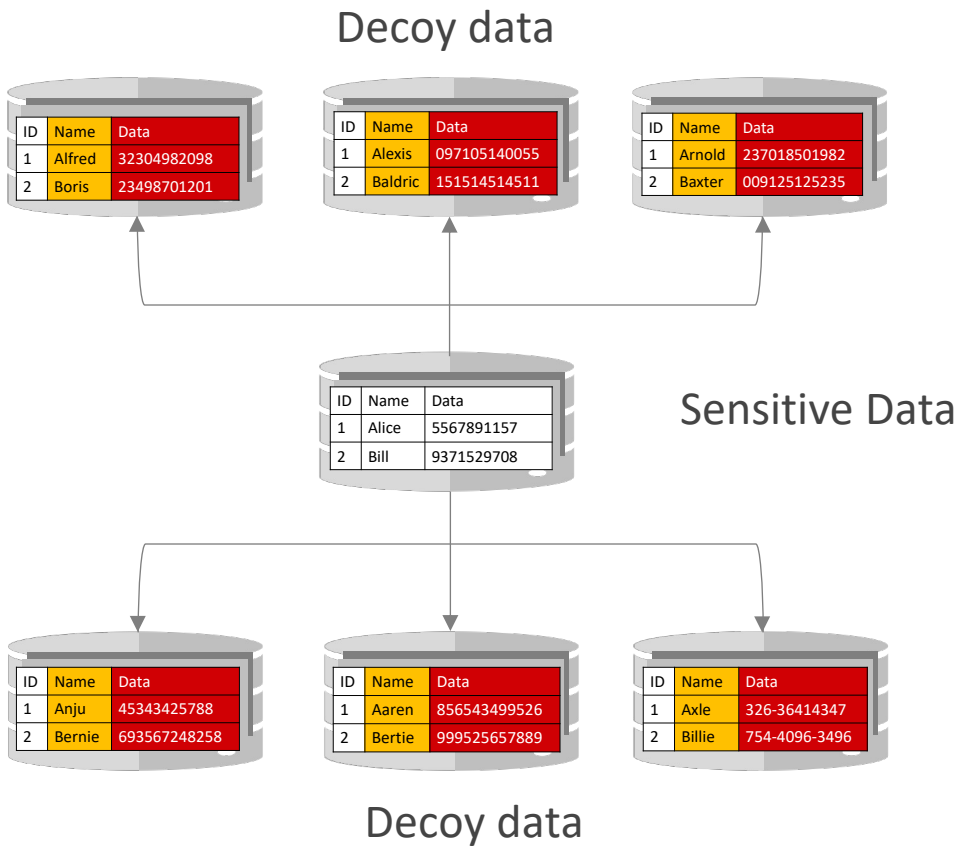
Use case #7: Decoy Data

Problem

- As part of a defense in depth strategy, organization create decoy databases containing seemingly valid data
- Ongoing need to create new decoys, potentially at short notice whilst repelling an active attack
- Requirement to create honey pot datasets to attract potential attackers and learn about methods and tools

Solution

- Policies within DPM easyData used to create multiple decoy copies of production datasets
- Copies may be encrypted or contain apparently valid data
- Outcome: the ‘truth’ is disguised by a vanguard of decoys



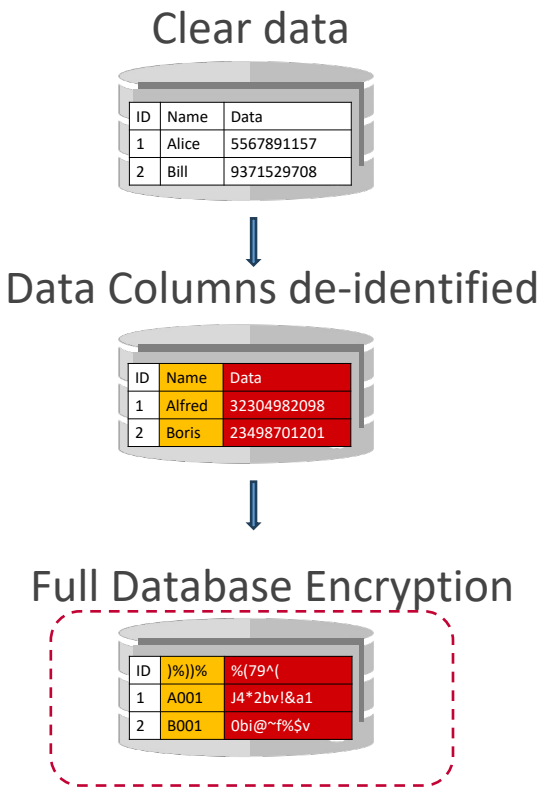
Use case #8: Multi-layer encryption protection

Problem

- Organizations seeking to thwart sophisticated attackers who potential possess advanced technical skills
- Sophisticated attackers may compromise some layers of protection and may be patient gathering information on internal systems for use in later attacks

Solution

- Obfuscate data protection by using multiple layers of encryption
- Utilize multiple encryption solutions such as protecting a database with a) column-level encryption via DPM easyData, and b) Transparent Data Encryption via DPM easyCipher and or a TDE solution provided by the database vendor



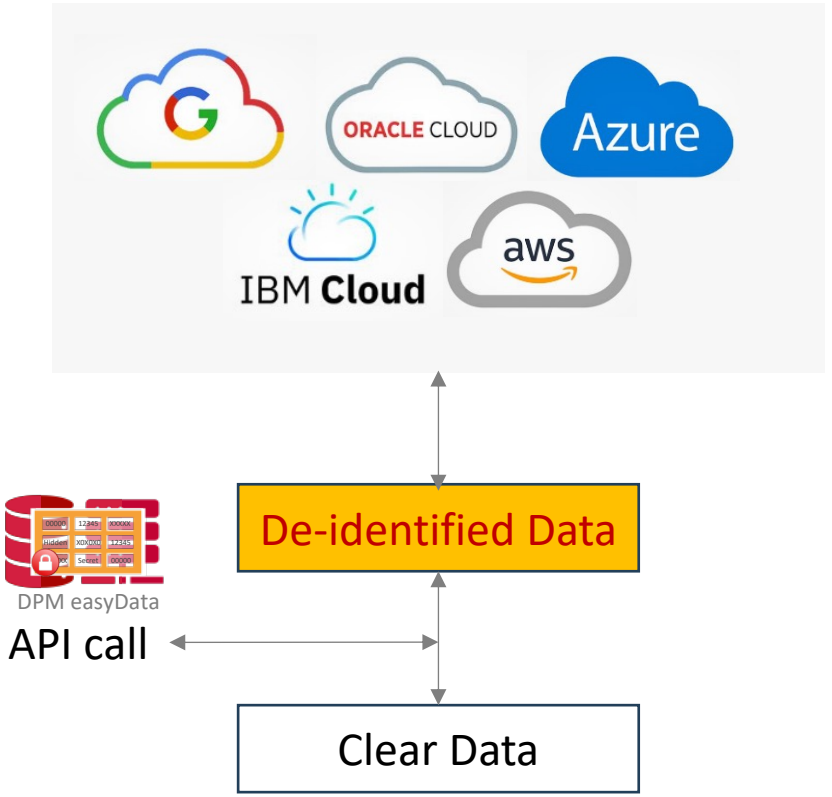
Use case #9: Cloud Data Protection

Problem

- Organizations wish to reduce costs by utilizing public cloud resources and outsource IT skills without compromising data security and controlling data sovereignty

Solution

- Code level data de-identification using DPM easyData
- De-identified data can be stored anywhere, on any system without risk of original data exfiltration
- Organization can use outsource IT resources, also without risk of original data exfiltration



Randtronics LLC

Milpitas CA 95035 United States
+1 (650) 241 2671
enquiry@randtronics.com

Randtronics Pty Limited

S1.1, Level 1, Building A 64 Talavera Road
North Ryde, NSW 2113 Australia
+61 418 226 234

Thank you for your time

email: bob.adhar@randtronics.com

Cell: +614 18 226 234 or +1 650 241 2671