PBD ELLIPTICO

Data Protection framework implementing privacy-by-design principles

INTRO

Commercial organizations have always tried to gain new markets and implement the most modern and disrupting technologies. As a driver of change, this can introduce major competitive advantages as well as various organizational and operational challenges.

Digitization is one of the typical examples of such a disruptive set of technologies affecting all industries - from Communications and Fin-tech to Government Services, it is an indispensable part of the modern life. Some of those industries, like Online Advertisement and E-commerce, have seen enormous growth while employing the discovery and utilization of a brand new resource – Data - “the oil of the 21st century”.

Within this document we will be looking into one of the less popular aspects of Data – privacy and sensitivity. While not immediately apparent, the importance of this aspect is becoming more and more relevant for companies, both due legislative as well as customer pressure. Examples of heavy GDPR fines, as well as market value crashes of breached companies, have significantly increased the urgency to implement a **data protection framework with privacy built in the organization foundations**

APPROACH

The following principles are taken from the Privacy-by-Design approach [1]. While some of them advocate Privacy to be integrated in the initial design, which is not a requirement for the Elliptico Framework, they share the same general spirit and approach.

Proactive not reactive; preventative not remedial

The privacy by design approach is characterized by proactive rather than reactive measures. It anticipates and prevents privacy invasive events before they happen. Privacy by design does not wait for privacy risks to materialize, nor does it offer remedies for resolving privacy infractions once they have occurred — it aims to prevent them from occurring. In short, privacy by design comes before-the-fact, not after.

Privacy as the default

Privacy by design seeks to deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given IT system or business practice. If an individual does nothing, their privacy still remains intact. No action is required on the part of the individual to protect their privacy — it is built into the system, by default.

Privacy embedded into design

Privacy by design is embedded into the design and architecture of IT systems as well as business practices. It is not bolted on as an add-on, after the fact. The result is that privacy becomes an essential component of the core functionality being delivered. Privacy is integral to the system without diminishing functionality.

Full functionality – positive-sum, not zero-sum

Privacy by design seeks to accommodate all legitimate interests and objectives in a positive-sum “win-win” manner, not through a dated, zero-sum approach, where unnecessary trade-offs are made. Privacy by design avoids the pretense of false dichotomies, such as privacy versus security, demonstrating that it is possible to have both.

End-to-end security – full lifecycle protection

Privacy by design, having been embedded into the system prior to the first element of information being collected, extends securely throughout the entire lifecycle of the data involved — strong security measures are essential to privacy, from start to finish. This ensures that all data are securely retained, and then securely destroyed at the end of the process, in a timely fashion. Thus, privacy by design ensures cradle-to-grave, secure lifecycle management of information, end-to-end.

Visibility and transparency – keep it open

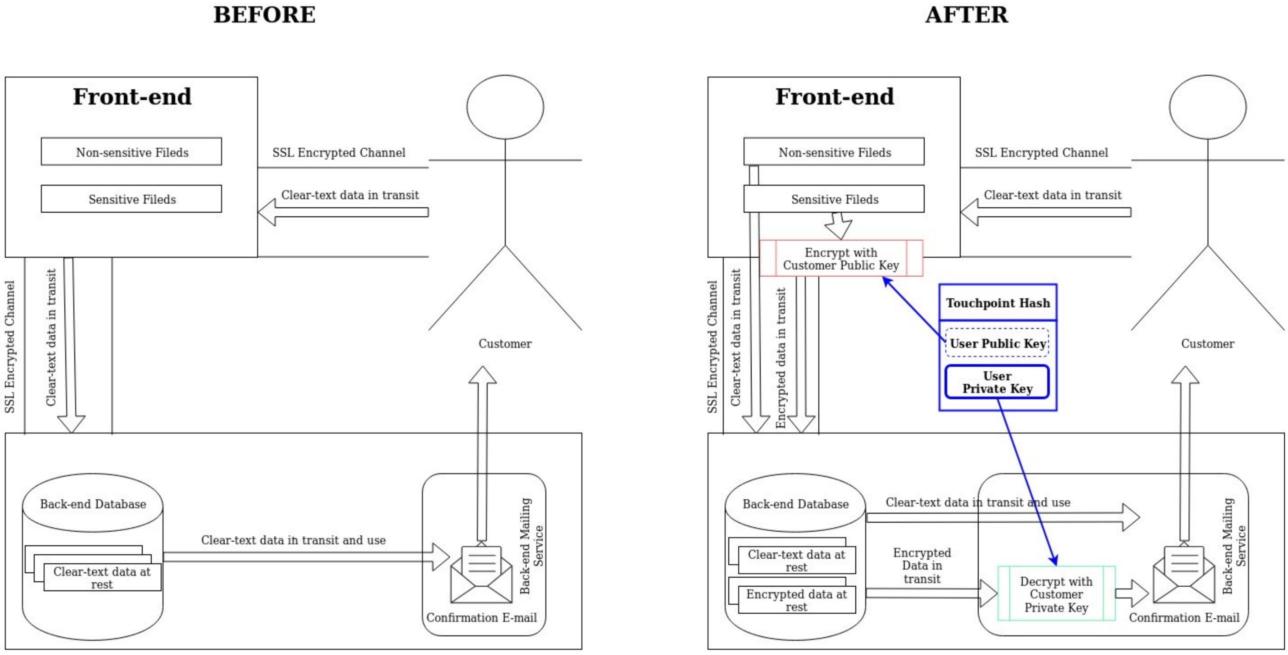
Privacy by design seeks to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification. Its component parts and operations remain visible and transparent, to users and providers alike. Remember, trust but verify.

Respect for user privacy – keep it user-centric

Above all, privacy by design requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options. Keep it user-centric.

SOLUTION

PbD Elliptico is an end-to-end data encryption and access management framework that is aimed at covering the current GDPR[2] requirements and at mitigating privacy and security risks. By applying a robust encryption philosophy and transparent key-management facilities, it elegantly solves the most common scenarios faced by companies, while dealing with sensitive user data.



In the above example, we are demonstrating a highly simplified operations model, where the Customer is placing an Order via the Company Front-end and needs to receive an order confirmation via email.

A very high-level explanation shows how Elliptico gets plugged into the process:

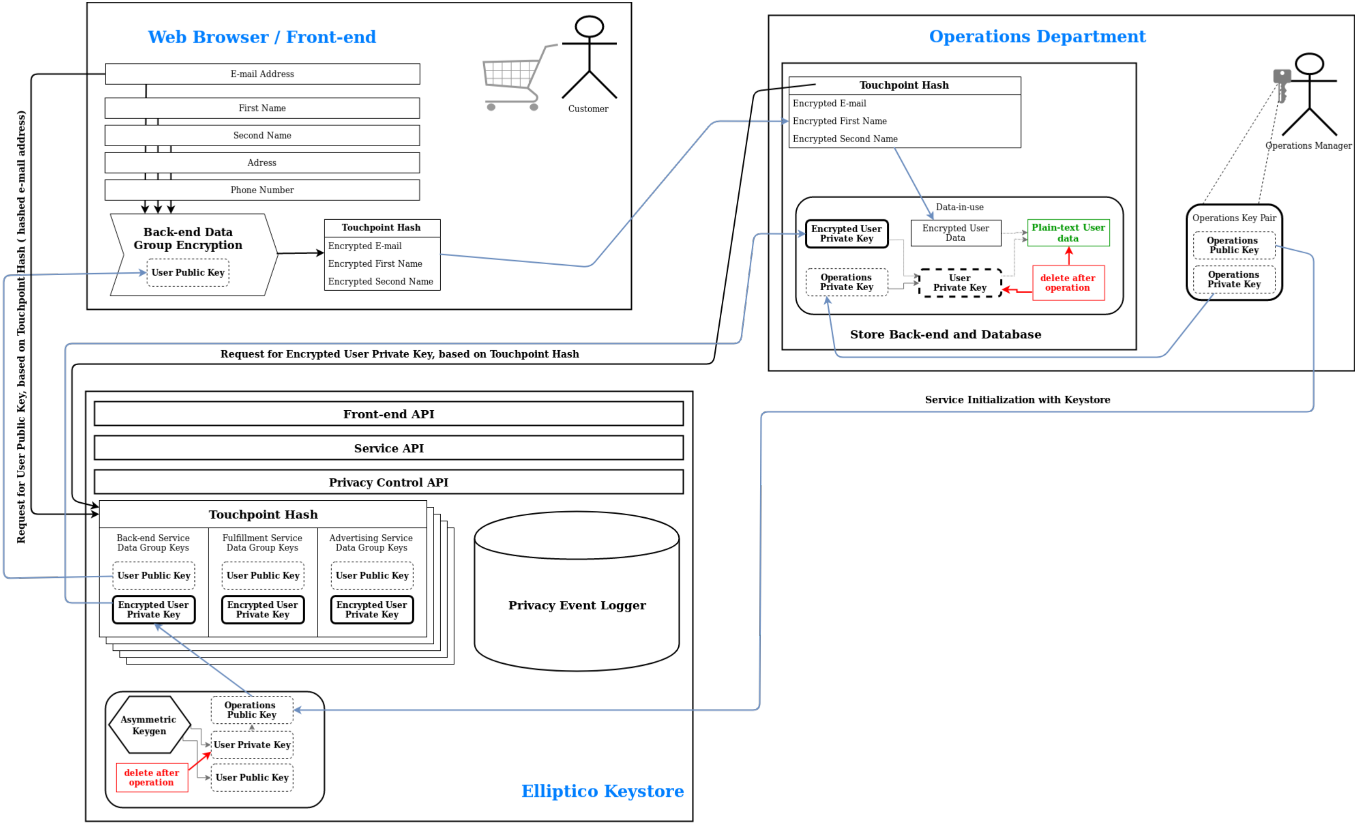
- Customer is identified by a Touchpoint Hash generated by the Front-end to get a Customer Public key from Elliptico and to encrypt all the sensitive fields,
- The Back-end receives and stores encrypted data,
- The mailing service, based on the Touchpoint Hash requests the Customer Private Key [*] and decrypts the required sensitive fields.

This approach gives multiple advantages:

- Data in Transit and at Rest is encrypted [3]
- Only Data in Use is decrypted and only for the predefined uses
- Customer Private Keys are protected and can’t be stored permanently
- Data Warehousing solutions can still use sensitive data for table linking, but can’t decrypt it

SINGLE SERVICE EXAMPLE

The following example extends the previously shown scenario with more details on Key distribution and management:



1. The Operations Manager sets-up the Framework by defining the Sensitive Data Groups and the Touchpoint Hash. He also generates a dedicated Operations Public-Private key pair.

2. He sends the Operations Public Key to the Keystore and securely stores the Operations Private Key within the back-end.

3. A customer is ready to finalize an order within the Front-End. He starts filling his e-mail address and other sensitive data. As soon as the e-mail is complete it is hashed and is sent to the keystore with a request for a dedicated Customer Public Key.

4. Within the Front-end, the browser will use the received Customer Public Key to one-by-one encrypt all the sensitive fields and will send them over to the Back-end in the same way it sends non-sensitive data. The Back-end will store the data and wait for the next step.

5. When the Order is complete, the Back-end will have to send a confirmation e-mail, for which it will have to decrypt the E-mail address and name of the customer. It starts by sending a request for the Encrypted Customer Private key using the Touchpoint Hash as an identifier.

6. The Keystore wills send back the Encrypted Customer Private key, which can then be decrypted within the Back-end using the Operations Service Private Key.

7. The decrypted Customer Private Key is then used to decrypt the customer e-mail and names. After the e-mail is sent, the sensitive data is no longer needed, so the Back-end deletes both the decrypted Customer Private Key as well as the decrypted Customer Data.

DETAILED MULTI-SERVICE EXAMPLE

INITIAL SETUP

SENSITIVE DATA GROUPS

As required by GDPR each sensitive customer data parameter needs to be collected for a specific purpose and its storage and usage traced in the company and partner systems. This means that at the beginning of the operations setup, the company is required to consider all the sensitive customer data it needs and group it by the usage and consent required. The company also collects many sets of non-sensitive data points – those are operated in a classic way and are outside of the scope of this framework.

Example of sensitive data groups by the purpose:

Web shop operations (to be able to receive order confirmation, check order status, cancel, trace, etc.)
E-mail address,
First Name,
Last Name

Fulfillment (deliver the order to the actual customer address)
First Name,
Last Name,
E-mail,
Address,
Phone Number

Advertising (sending targeted product listings, banners, etc.)
E-mail
Age
Gender
Country
Facebook Profile Link

CUSTOMER TOUCHPOINTS

TOUCHPOINT CHOICE

The shop needs to decide which customer data parameter will be used for its identification. In some companies this may be a significant problem if they have many different customer entry points – this can still be solved with the current framework, but will be the topic of a different document. In the current example, the shop has only one entry point – the Shop Front-End and it always requires e-mail based login for any purchase operation. Just browsing the store may not require the customer to identify, but to make an order or check status he needs to provide an e-mail.

TOUCHPOINT HASHING

Customer Touchpoint - e-mail address –
john.doe@protonmail.ch
Hashed Customer Touchpont – SHA256(e-mail address) -
f7766a01f208327161eef7dfe32aee475b532c5fb3d847ddab4fd081dc1e980e

The SHA256 function is implementing a robust hashing algorithm, that is very hard and expensive to reverse, while giving the same result, regardless of the platform and the specific implementation as long as the input string is the same:

```
echo 'john.doe@protonmail.ch' | sha256sum
f7766a01f208327161eef7dfe32aee475b532c5fb3d847ddab4fd081dc1e980e  -
```

It is does not to allow hash collisions and in need can easily be replaced with an even higher resolution hash function without a significant effort.

TOUCHPOINT TRANSIT AND STORAGE PRINCIPLES

The hashed customer touchpoint can’t be reversed back to the actual e-mail address. It is just a unique identifier, that is safe to transfer, store and share. The actual non-hashed touchpoint data should never leave Web Browser of the Customer. In case the data used for the touchpoint is in fact needed for a specific purpose, like sending an advertising e-mail or package tracking information, it is stored separate as an Encrypted Dataset.

SERVICE KEYPAIR GENERATION

A public-private keypair needs to be generated in advance for each Service, i.e. each Sensitive Data Group.

The Public Key of this pair will be stored within the Elliptico Keystore and used for encrypting the private key of each Customer Keypair.

The Private Key of this pair will be owned by the relevant department, stored in the services that will locally be decrypting and using the sensitive data.

ELLIPTICO KEYSTORE SETUP

For the initial setup, Elliptico Keystore needs to have each of the Sensitive Data Groups configured. Required parameters are:

- Data Group Name – arbitrary group or Service name
- Data Group Public Key or Service Public Key – the Service Public Key generated in the previous step, used to encrypt each of the Customer Private Keys for that Service

For this example the ServiceKeys table will look like:

Service Name	**Service Public Key**	**Expiration Period**
Operations	MF0wEwYHKoZIzj0CAQYIKoZIzj0DABADRgAE0ryG7Wg+78NMh1aqBw717Oh6sOSykLtbEi47RfrJ96+rt1+LiXueLuTIi4dVjIDVOxmtl1dm1ew76nBXz88EipB/6rA=	12 months
Fulfillment	MF0wEwYHKoZIzj0CAQYIKoZIzj0DABADRgAE3zskX8MCMh2RxicA4P+J3Ky+F3aClHYgli71Q8kxjUYC/v9n4smaXtJFwTQrloUgEC8iQ4H7Ld4s84RbUJMHphZjb8k=	30 days
Advertising	MF0wEwYHKoZIzj0CAQYIKoZIzj0DABADRgAEMQohdAVZwkLTeZt1kCW5teHCjv3khT0+Ji4VCRPQ2QnEpLzjZKZxGfkzCpEagd7RsGFXiCLX8GjQqoXuQgN58xLBvaE=	12 months

SHOP FRONT-END SETUP

Each of the sensitive fields that will be required from the customer needs to be identified and coded in a specific way. A typical Web front-end, running for example in JavaScript, will be collecting and verifying all the different fields locally, within the Browser isolated process and will be sending them to the Back-end one-by-one on action, or en-mass with a “Submit” button.

The change required for the current implementation scenario will require:

- On e-mail complete & verified, run a SHA256 function on the e-mail and send the result to the Elliptico keystore API, which should reply with a set of dedicated customer public keys. (one for each service)

- adding an encryption function to each of the fields before sending it to the Back-End, using the received public keys
- Sending only the encrypted fields and never sending or storing the non-encrypted sensitive fields outside the Web Browser Process.
- **Advertising scripts should not be allowed access to the raw non-encrypted sensitive fields**

SERVICES SETUP

Services not allowed to use the sensitive data:
Since the Shop back-end does not need to use the decrypted data, there are no modifications required for it, other that accommodating the new field size after encryption. Same goes for Datawarehousing, Analytical, Backup, Cashing, etc. services.

Services that will use the sensitive data:

- They need to be able to securely store the Private Service key
- They need to be able to request the Encrypted Customer Private key from the Elliptico keystore, based on the touchpoint hash
- They should be able to decrypt the Customer Private key in a secure way and should never store that key longer than needed
- They should be able to decrypt the sensitive customer data using the Customer Private key and should never cash is or store it for longer than needed

The above listed changes will require some rewriting/refactoring of the service software and implementing some principles that are anyways required by the GDPR and are considered good security practices.

SHOPPING EXPERIENCE PHASE

The customer visits the online Shop and creates an order

1. The Customer visits the online Shop using a web browser. No login is required to browse the products and add them to a shopping cart.
2. Once the Customer is ready he goes to “Check-out” and loads the Front-End Checkout page in the browser. He gets a number of fields to fill up and starts filling them one by one. Fields are not submitted until the correct order of actions is complete.
3. The first action is to fill the e-mail address. After local format verification, the it’s hashed with a SHA256 function and the result is sent to the Elliptico keystore.
4. Elliptico will then check the Customer Keys table and will try see if this touchpoint is already known or not. If known, it will respond with the set of Customer Public keys – one for each configured service. If not, it will assign the touchpoint hash to a set of non-taken Customer Public keys and will again send them back to the customer web browser.
5. The web-browser will store the keys in memory and will start using them

for encryption of the sensitive fields.

6. After the user fills each of the sensitive fields, it will be encrypted with the relevant Customer Public Key and can be then stored or sent to the back-end or directly to the service that needs them.
7. In the case of the e-mail, it will be encrypted with each of the service keys, as it is needed for all three of the services. All three of the encrypted versions of the e-mails should be then stored or sent to the back-end or the relevant services together with the customer touchpoint hash. In our example two of the encrypted versions go to the back-end and one is sent directly to the advertising service.
8. The customer completes the form, sends all the encrypted sensitive fields and other non-encrypted fields and finishes the interaction with the Shop Front-end.

OPERATIONS PHASE

The back-end needs to send an order confirmation e-mail

1. The back-end has received the customer touchpoint hash together with the encrypted e-mail address of the customer.
2. The back-end compiles an order confirmation with the non-sensitive product information, order number and customer touchpoint.
3. The back-end will then send a request for the Operations Customer Private Key to the Keystore together with the customer touchpoint hash.
4. The Keystore will log the request and will return the encrypted Operations Customer Private Key
5. The back-end, representing the Operations Service will have the Operations Service Private key stored locally in a secure way. It will use it to decrypt the Customer Private Key and will store it locally in a secure way.
6. The back-end can then decrypt the customer e-mail address and send the required order confirmation e-mail
7. After the e-mail is sent, the back-end shall delete the Customer Private Key and the decrypted e-mail address from its memory and storage.
8. The encrypted e-mail address is still available in the safe form to be used in data warehousing, analytics or other services

FULFILLMENT PHASE

The fulfillment department, needs to deliver the ordered goods to the physical address of the customer.

1. From the back-end, the fulfillment service receives the non-sensitive order information, the customer touchpoint hash and the encrypted sensitive fields that are dedicated to it
2. The fulfillment department prepares the order and needs to share the delivery address and customer contact information with the Delivery Company.
3. They will then send a request for the Fulfillment Customer Private Key to

the Keystore together with the customer touchpoint hash.

4. The Keystore will log the request and will return the encrypted Fulfillment Customer Private Key
5. The fulfillment department will have the Operations Service Private key stored locally in a secure way. It will use it to decrypt the Customer Private Key and will store it locally in a secure way.
6. The fulfillment department can then decrypt the customer e-mail, name, address, phone number, etc sensitive information needed for the order delivery. It can share them with a Delivery company and also send delivery confirmation e-mail

ADVERTISING PHASE

The advertising department needs to send a newsletter to the customers who have opted-in based on their previous orders and location information

1. Advertising service has received the non-sensitive order information, the customer touchpoint hash and the encrypted sensitive customer parameters like e-mail, age, gender, location, facebook profile etc.
2. Dependent on the analysis approach the Advertising service may or may not need to decrypt all the sensitive parameter to prepare a targeted advertising. Small enumerations like gender, age, country, etc may need extra salting in order to achieve proper encryption. To decrypt them, the Advertising service needs to send a request to the keystore, with the customer touchpoint hash and to request the Advertisement Customer Private Key.
3. Location use, may require sending the location data to third party for classification, which is only allowed without any other identifying information. The same rules apply for that as for the other 3rd party services.
4. The keystore will log the request and respond with the encrypted Advertisement Customer Private Key.
5. The Advertisement service, configured to securely store the Advertisement Service Private Key will then be able to decrypt and securely and temporary store the Advertisement Customer Private Key.
6. It can then decrypt all the sensitive Advertisement Customer data, including the e-mail address and generate the Advertisement newsletter
7. All decrypted sensitive fields and the Customer Private Key should then be deleted and if needed again the process should repeat.

EXPIRATION PHASE

According to GDPR sensitive data should be removed after a disclosed period of time. With the Elliptico framework the actual data is not removed, but it is made unreadable by disposal of the private keys which can decrypt it.

1. The Elliptico keystore keeps a detailed log of all public and private key access requests. Each public key access request overwrites the last customer access date.
2. The keystore constantly monitors if the expiration period for a certain key record is reached and if that happens the expired private key is deleted

from the table
3. Once deleted, no service will be able to request the Customer Private Key and will no longer be able to decrypt the data.
4. The encrypted data for that customer and service is still available in all multiple locations, but the sensitive information can no longer be extracted or recovered.
5. In case the same customer (touchpoint hash) is later reintroduced, the keystore will assign new key pair and all the old data will stay unreachable.

RIGHT TO BE FORGOTTEN SCENARIO

Each customer affected by GDPR has the right to request the removal of his private data from a company as soon as their business is complete.

1. Once the customer has finished business with the Shop and has decided he no longer wants to share his sensitive information with it, he is able to request the deletion of a Customer Private Key.
2. Different methods of customer authentication can be used to confirm the identity and remove the possibility of malicious Private Key delete requests. An Elliptico password may be a good way to allow the user more control over his keys, as well as access to usage logs and expiration dates.

EXTERNAL REFERENCES

[1] *Cavoukian, Ann. "7 Foundational Principles" (PDF).*
[2] *"EUR-Lex – 32016R0679 – EN – EUR-Lex". eur-lex.europa.eu*
[3] https://en.wikipedia.org/wiki/Data_at_rest

GLOSSARY OF ACTORS AND TERMINOLOGY

Shop (Company) – the e-commerce company operating the business in focus

Customer – the party buying the goods provided by the company

Shop Front-end – A web based interface, that drives the customer experience, is ran in the Customer web browser and communicates with the Shop Back-end

Shop Back-end – An API based service that provides product and inventory information, supports the order process as well as other Front-End functions. This service is in charge of initially receiving and storing user provided data and will forward it to the specific departments that need to use it

Shop Analytical Back-End – A data gathering, integration and visualization tooling that is providing off-line data gathering, analytical and reporting facilities to the Shop Sales and Accounting departments.

Operations Department – This department is in charge of the Shop Front and

Backend, sending order confirmations, etc.

Accounting department – the department in charge of accounting, compliance and auditing

Fulfillment department – the department in charge of fulfilling the customer purchase as well as facilitating returns

Customer support department - department in charge of customer support with Call, Mail and Chat service

Elliptico Key store – The core framework service providing an API for generation, distribution and audit of customer and service keys.

Customer Touchpoint – a customer provided data parameter, that is used for general user identification within the Shop front-end. A typical example is login name or e-mail address.

Customer Touchpoint Hash – since the customer touchpoint can be sensitive information it self, it is stored as a result of a mathematical hash function, which will always return the same result for the same touchpoint data

Customer Key pair – a pair of public – private keys that are generated for each customer Touchpoint. Each customer may have several pairs, dependent on the number of Data Collection Purposes

Sensitive Data Group – a group of sensitive data parameters that are collected to be used for a specific purpose that is disclosed in advance and typically accessed and by a dedicated Company Service

Company Service - a service provided by the company that abstracts a group of similar operations. For example Fulfillment, Delivery, Sales, etc.

Service Key pair – a pair of keys generated for each service or department, used to encrypt the Private Keys in the Customer Key pairs

Purpose Data collection Rule – Data should only be collected for a specific purpose that is disclosed to the customer in advance. Data can’t be collected for one purpose and then reused for another without the customer consent

Right To be forgotten – every customer has the right to request deletion of his personal information in case he no-longer needs to use the service and revokes his consent of the data usage. That would mean deletion of the data from all company and partner systems and services.