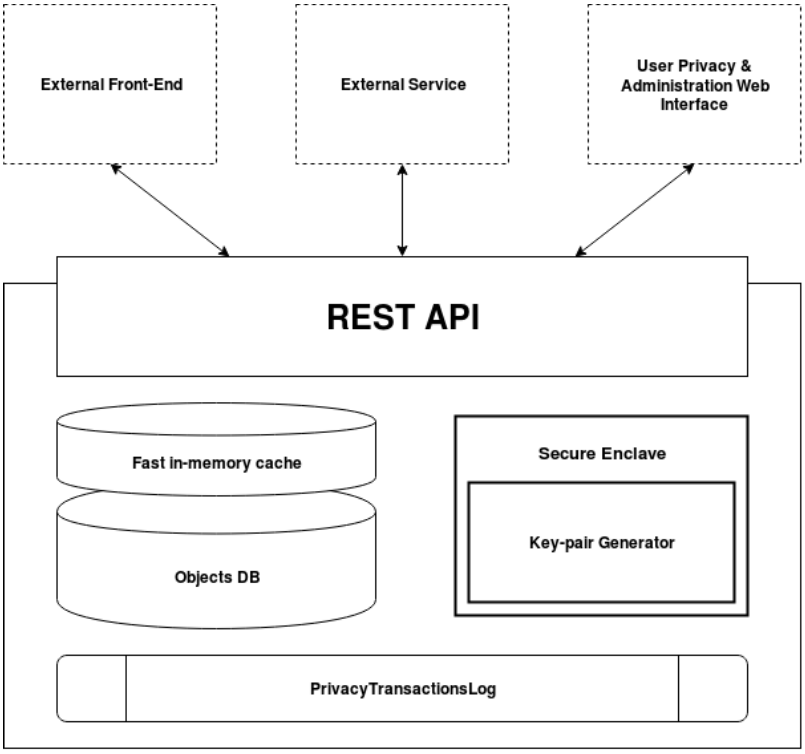
PbD Elliptico Keystore Architecture

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 private and sensitive data scenarios.

High-level Design

The Key-store design is based on a generic RESTfull API implementation with an Object Database, Transactional Logging and an integrated Secure Enclave for the key-pair generation.



REST API – flexible SSL enabled web-service interface that can serve any External Front-end, Service, User privacy or Administration Interface

Object DB – SQLAlchemy supported Objects Database (pluggable)

Fast In-memory cache – Objects cache to reduce request response times

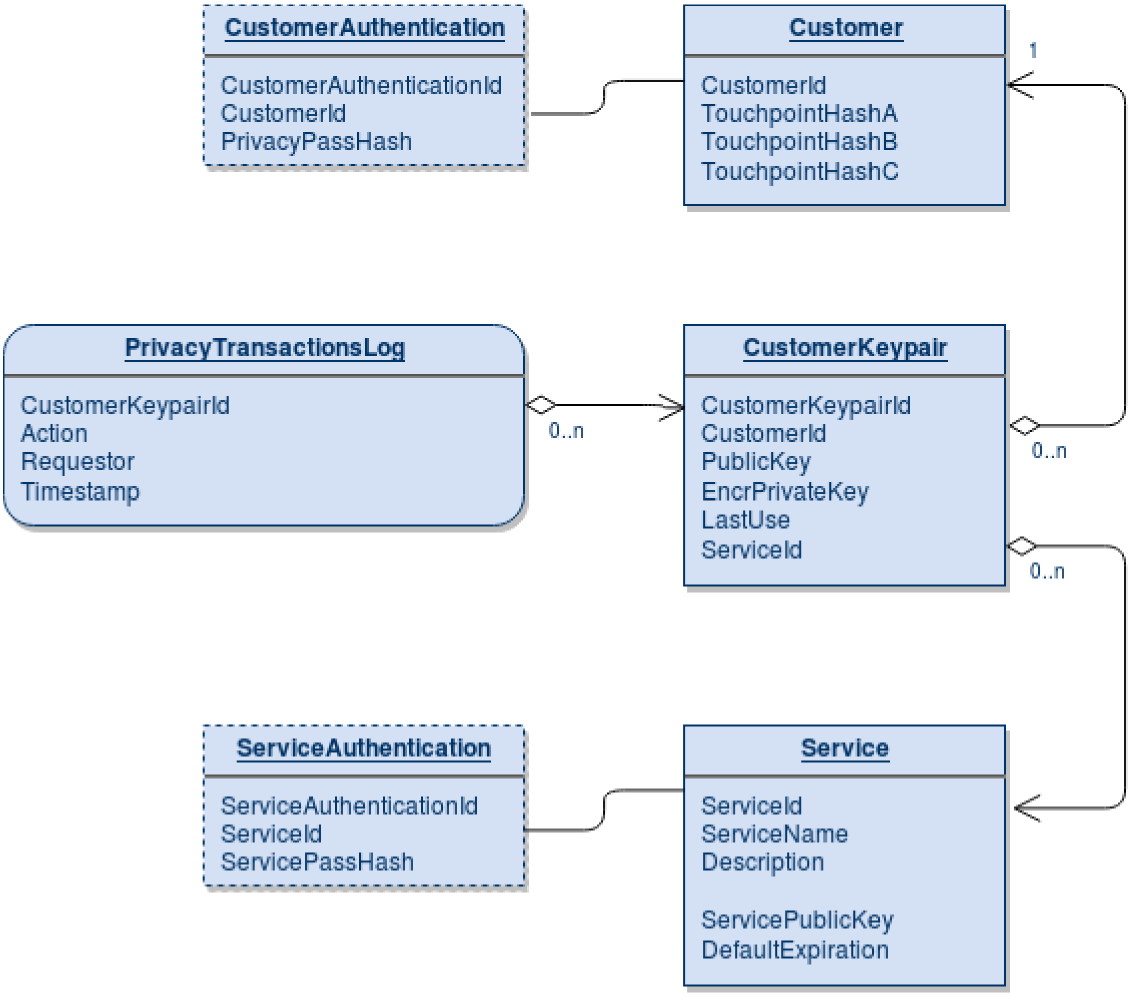
Secure Enclave – A dedicated hardware or software module that is well insulated from the rest of the key-store

Key-pair Generator – Plug-in style key-pair generator that can efficiently generate RSA, Elliptic Curve or other Key-pairs for service per customer

Privacy Transactions Log – an immutable log that stores all API, Cache and Database interactions

Object Model

Elliptico Key-store is following some base Object Oriented Programming principles. The following diagram shows the high-level objects, properties and their relations as they will be stored in the Key-store Database.



- **Customer** – this class describes the Customer and stores one or more Touchpoint Hashes.
- **CustomerKeypair** – class describing and storing each of the service key-pairs unique to the customer. Typically each customer will have one key-pair per service.
- **Service** – the class describing the Service that will be using the sensitive customer data. Stores the Service Public Key. The Service Private Key is never available in the key-store.
- **Service & Customer Authentication** – non-mandatory classes that can allow the Customer or the Service Representative to be authenticated and to expire keys or monitor privacy transactions
- **PrivacyTransactionsLog** – while not a real class, this Log table stores all actions done with the Customer Keypairs – key requests, expirations, etc.

Key Design Principles [Draft]

- The Elliptico Keystore does not store the Customer Private Keys. They are generated within the secure enclave, encrypted and their plain-text version is securely removed
- The Secure Enclave has limited memory assigned to further limit any possibility for Private Key Leaks
- The Key-pair generator uses a high-entropy random number generator, enhanced with all available random events to ensure proper key security
- The Keystore never handles, receives or stores the Service Private Key, so it is not able to decrypt the Customer Private Keys
- Key-pair generation is done out-of-band to the Customer Key requests, to ensure quick-response and smooth customer experience
- All Key request operations, Database actions, Cache requests, API calls are stored in the immutable log table to ensure