Simplifying Application Level Encryption:

An Overview of Baffle's Architecture and Configuration Modes

White Paper

June 2019

Baffle Example Architecture and Configuration Overview

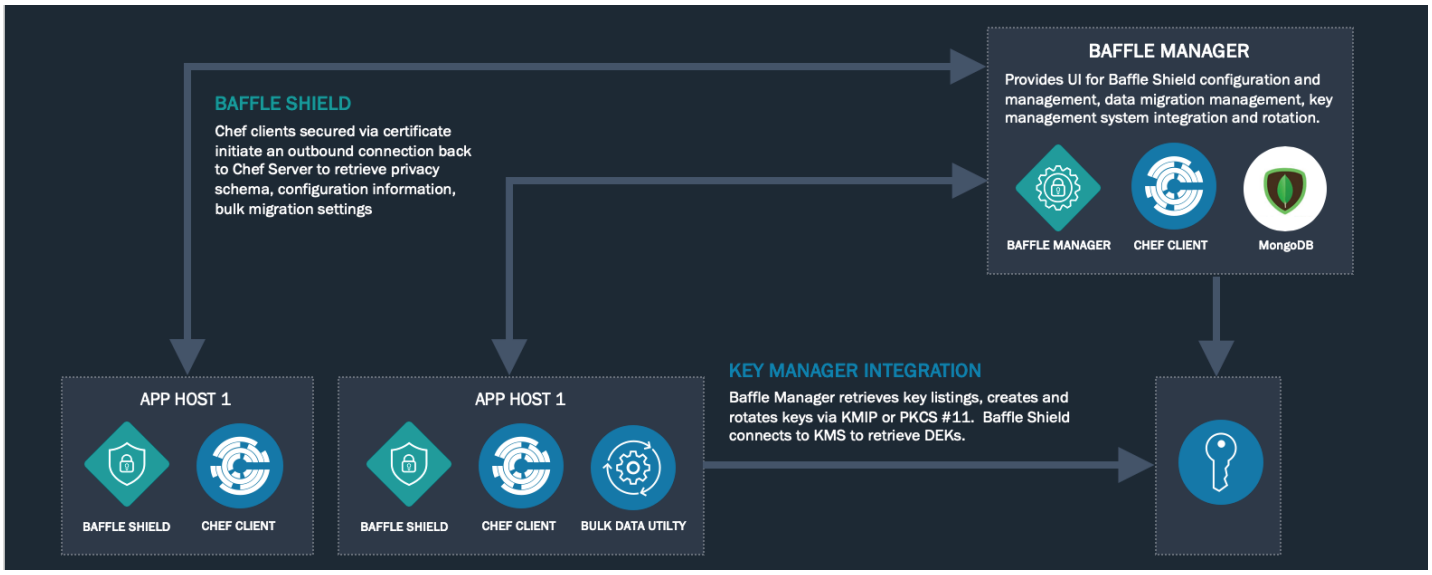
This document provides a high-level example architecture for Baffle deployment options and an explanation of configuration modes. More detailed information can be provided in consultation with a security architect.

Baffle's Advanced Data Protection consists of three main components:

1. *Baffle Manager* is the administrative console for the solution that integrates with enterprise key managers, databases and manages the Baffle solution components
2. *Baffle Shield* is the SQL / NOSQL proxy that functions to encrypt and decrypt data at the field or record level.
3. *Baffle Secure Multiparty Compute (SMPC)* is an optional component consisting of stateless servlets that enable secure computation on encrypted data such as sort, search, wildcard search and mathematical operations without ever decrypting the underlying values.

Sample Architecture:

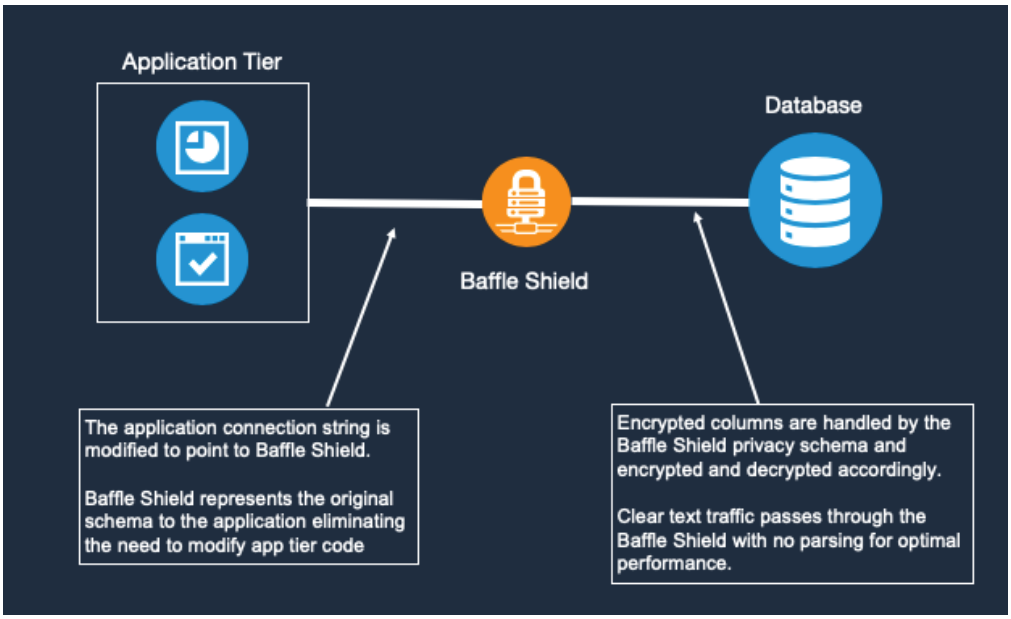
The diagram below shows the general communication flow amongst the different Baffle components and integration points. The Baffle Manager communicates with a Key Manager to establish a privacy schema for the application that represents the original data schema to the application tier while mapping encryption keys to respective fields. The mapping is communicated to the Baffle Shield which, in turn, retrieves keys in a stateless manner and uses them for encryption/decryption. The SMPC servlets are not represented in this diagram.



Communication with key stores is enabled via industry standard protocols leveraging KMIP v1.1 or higher for key managers and a PKCS #11 library to interface with hardware security modules (HSMs). For cloud key managers such as AWS KMS and Azure Key Vault, a REST API communication method is used.

Data encryption keys (DEKs) are encrypted with a master key and Baffle supports key rotation and multiple key versions. Encryption keys are never persisted in Baffle Manager and are only held in memory on Baffle Shield or in SMPC.

The diagram below shows the Baffle Shield placement within a traditional application and database architecture. Baffle Shield can also function with fat client and API-based access architectures such as microservices or serverless PaaS.



For configuration of the application tier, the application connection string is modified to point to Baffle Shield instead of the database, or alternatively, a DNS change can be implemented to route traffic to Baffle Shield. BaffleShield represents the original schema to the application eliminating the need to modify app tier code.

Encrypted columns are handled by the Baffle Shield privacy schema and encrypted and decrypted accordingly. Clear text traffic passes through the Baffle Shield with no parsing for optimal performance.

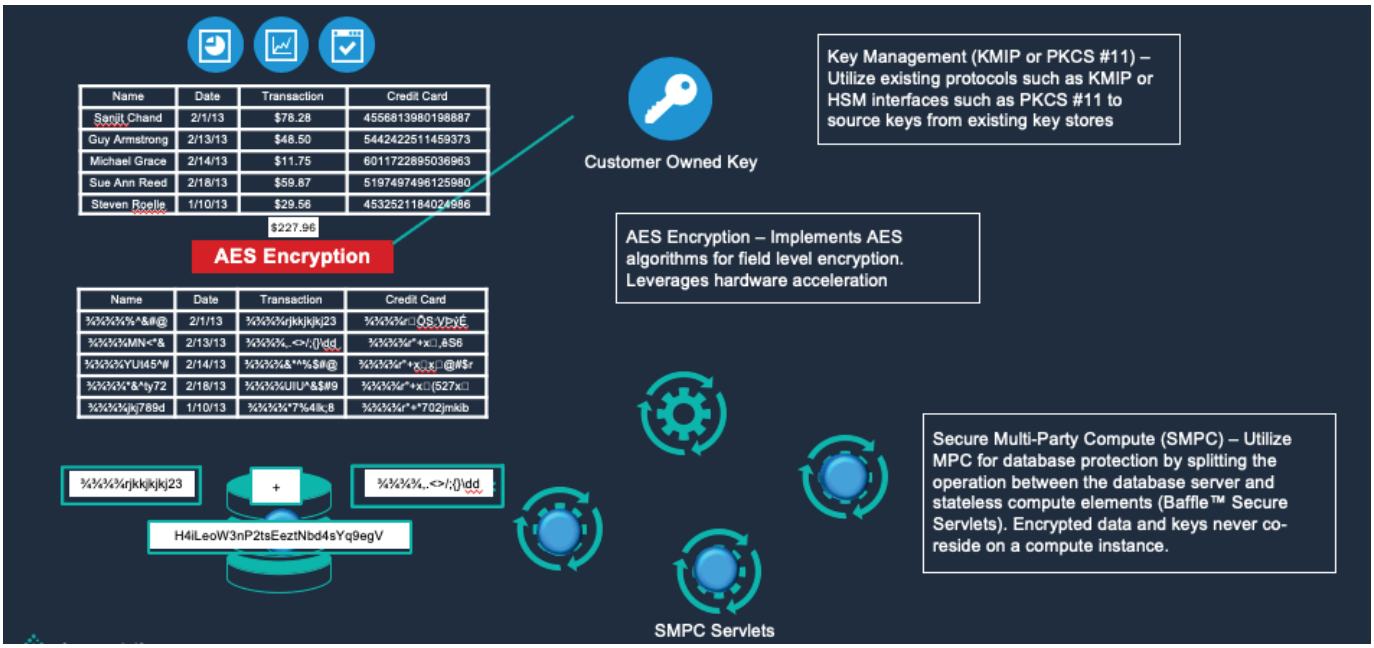
Baffle Shield can also be deployed on the application tier or behind load balancers to enable HA and higher connection concurrency.

Configuration Overview:

Baffle can be configured in four main modes as described below. All encryption modes use AES as the encryption algorithm.

1. ***Standard Encryption:*** Baffle functions as an application level encryption (ALE) equivalent in this mode encrypting data on a field level basis. This is performed using Baffle Manager as described above to enumerate the data schema and enable an encryption key mapping.
2. ***Record Level Encryption:*** Baffle can be configured for record level encryption to support multiple keys within a single column that are mapped to respective data owners or entities. This mode of encryption can be used effectively in multi-tenant or shared data environments where segmenting data can be difficult. In this mode, data shredding can be enabled by deleting keys for a respective entity
3. ***Data Masking:*** Baffle can enable simplified data masking to prevent decryption of data based on configuration or deleted keys. This mode can be used to minimize data exposure in test/dev environments and to better control data exfiltration to external parties.

4. ***Advanced Encryption:*** Baffle can be configured to enable operations on encrypted data to support optimal application functionality and minimize breakage of business processes. This mode supports "homomorphic-like" operations on encrypted data but uses an AES encryption algorithm. Secure Multiparty Compute (SMPC) is the cryptographic technique that is utilized to enable this method. The method employs a security contract where encrypted values are never co-mingled with encryption keys but facilitates operations on encrypted data via a message passing protocol between a database and a separate SMPC compute domain. As such, the database functions as an encrypted data store with no key present and operations are performed in conjunction with the SMPC implementation. Consequently, the data is encrypted in memory in the data store and in process. The figure below describes the advanced encryption process that allows operations on encrypted data.



In conclusion, the Baffle Advanced Data Protection solution consists of Baffle Manager, the admin console, Baffle Shield, a SQL/NoSQL proxy and Baffle SMPC Servlets, an optional component that enables secure computation on encrypted data. It integrates with existing key stores via a PKCS#11 or KMIP interface. The solution protects data all the way up to a record level granularity and supports four modes of protection depending on the level of security desired. This solution aims to make encryption simple to adopt without disrupting existing application functionality.

For more information, please visit ***https://baffle.io*** or email us at **info@baffle.io.**