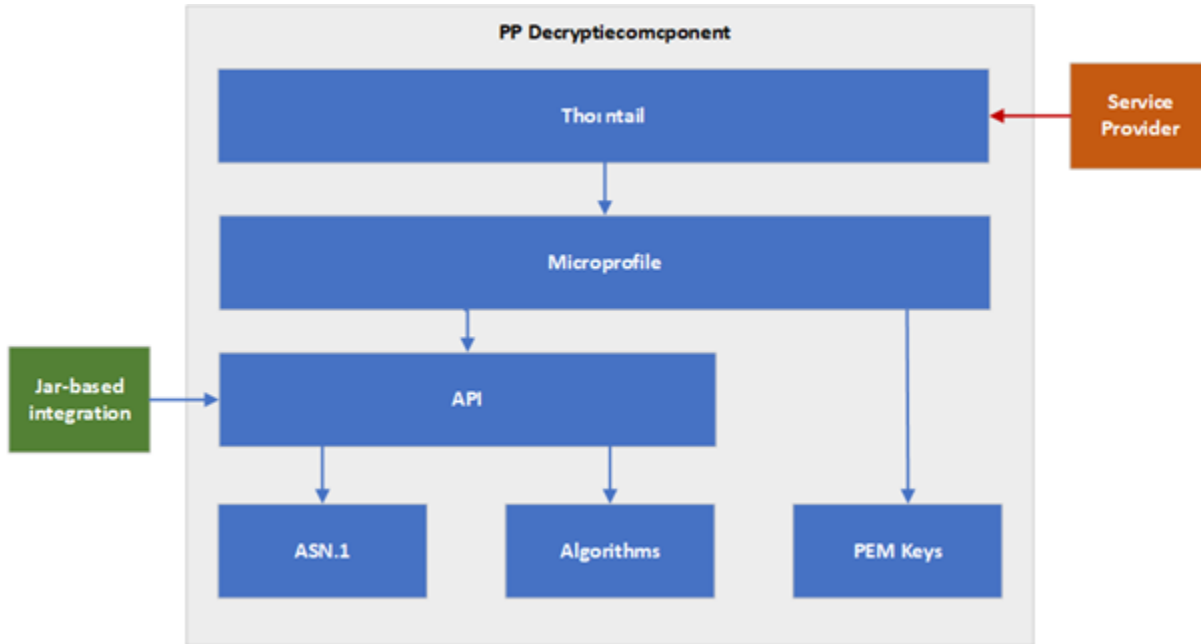


## 2a. BSNk Decryptiecomponent integration

The section outlines considerations regarding integration of the BSNk Decryptiecomponent. The BSNk Decryptiecomponent is provided in 2 ways:

- As a Docker container-image that offers a REST-based API
- Source-code, containing all functionality, offering possibilities for custom integration scenario's. (like Jar-based integration in the figure below)



### BSNk Decryptiecomponent integration considerations (Container)

Please consider the aspects below when integrating the BSNk Decryptiecomponent Container within your application environment

Aspect	Notes
Authentication and authorization	All requests to BSNk Decryptiecomponent are handled without authentication or authorization. Callers need to provide valid decryption-keys, schemekeys and valid encrypted structures.
Configuration	The container has no configuration settings itself. (zero-config) External connectivity must be configured through Docker. (See <a href="#">4b. Installation Decryptiecomponent (Docker-container)</a> , step 4 )
Connectivity (network)	The BSNk Decryptiecomponent's internal webserver operates at port 8080 and uses unsecured HTTP. Applications (and underlying infrastructure) that perform or support exchange of information with the container, must ensure the integrity and confidentiality of the exchanged data.  The BSNk Decryptiecomponent runs in isolation: it does not exchange information with external networks. It should not be accessible from a public network.
Key-management	The BSNk Decryptiecomponent does not support storage and management of decryption and conversion-keys. (BSNk-sleutelmateriaal). Key-management must be performed by calling application(s).*
Protocol / Service description	The container-image offers a REST-endpoint, that exposes the Decryptioncomponent functionality. Description of the API is provided as a Microprofile service description.

\* might be added in future versions

### Source-code

The BSNk Decryptiecomponent source-code is also provided.

The build process will yield the following output artifacts:

Artifact	Description
Decryptioncomponent JAR-files	Individual JAR-files, can be used for integrating BSNk Decryptiecomponent functionality into your own application.
Decryptioncomponent Thorntail application	This is the Java-EE application containing the REST-endpoint. (This application is also deployed in the BSNk Decryptiecomponent Container.)