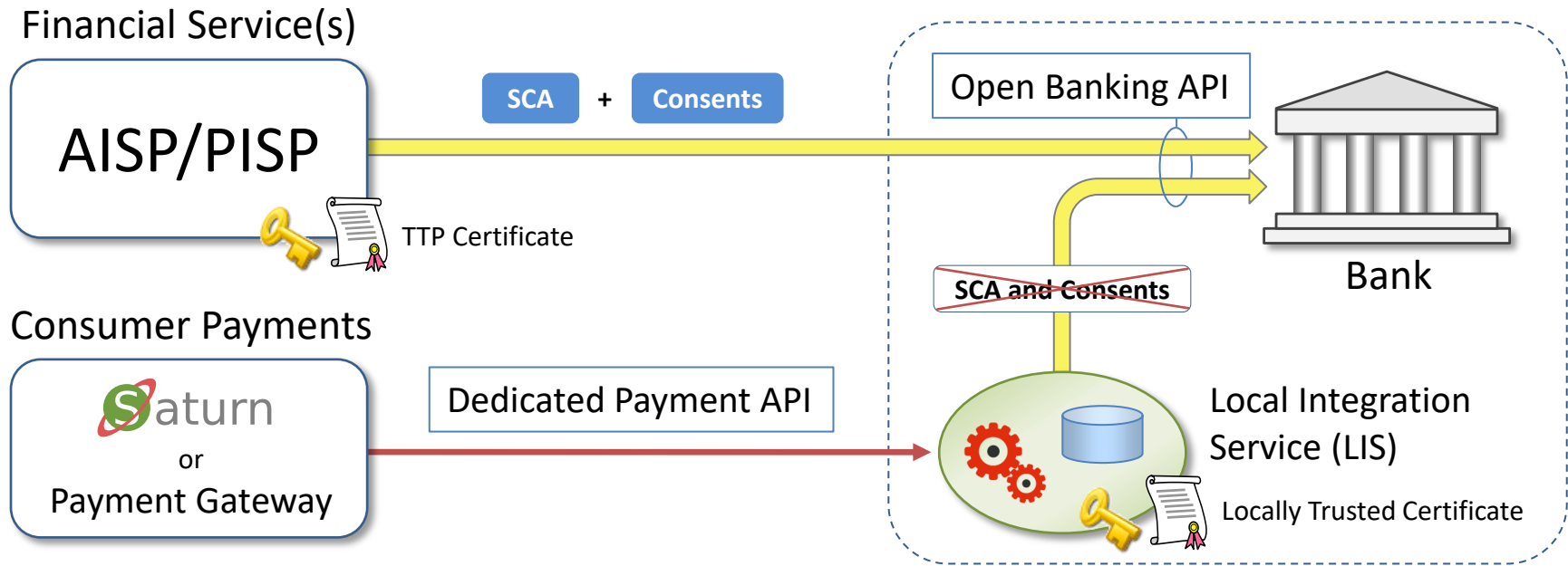


Dual Mode - Extending the Reach of Open Banking APIs



In theory Open Banking APIs can support *Consumer Payments*. However, the server centric OAuth security concept makes UX quite challenging compared to schemes like Apple Pay where SCA and account selection are integral parts of the “Wallet”.

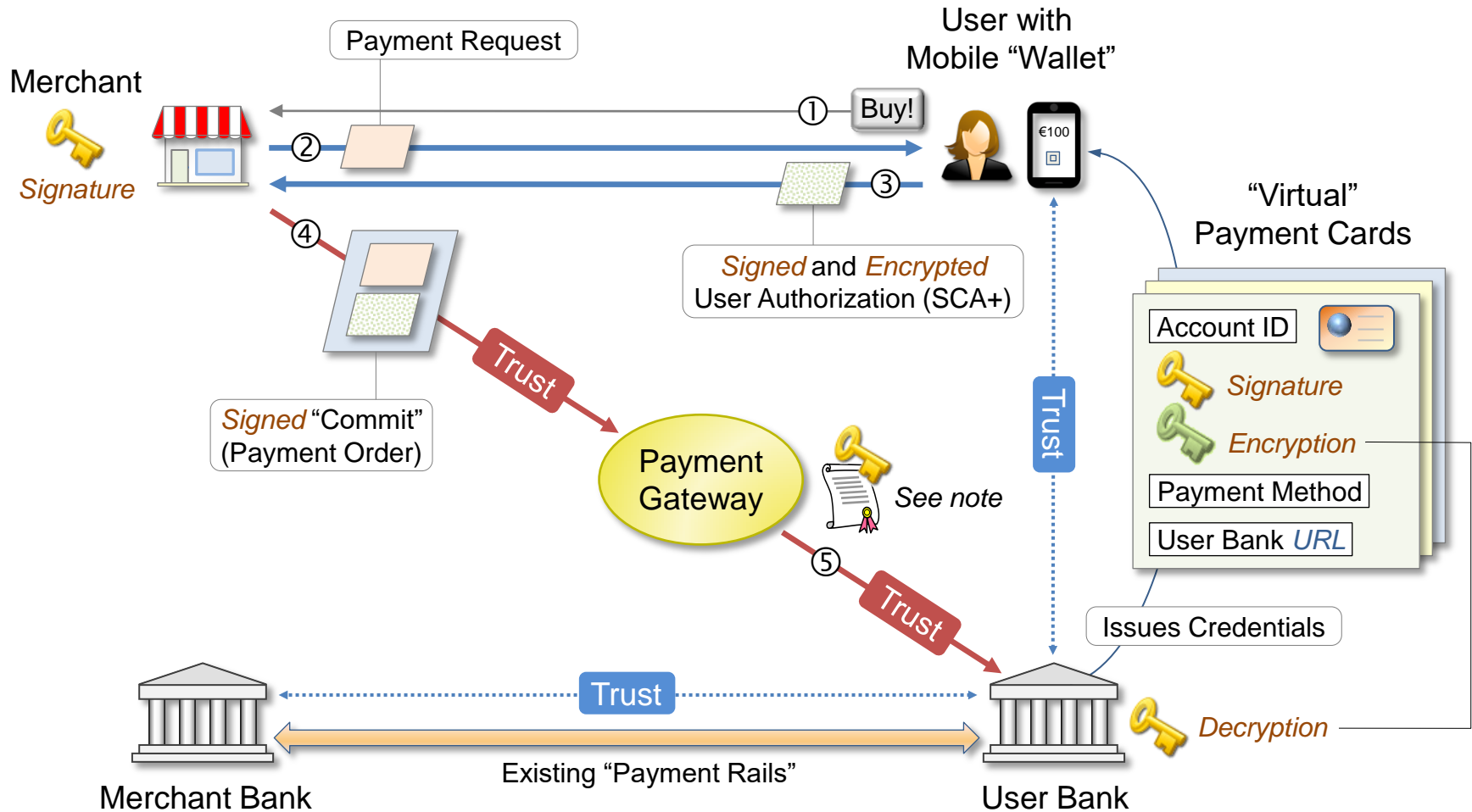
This presentation outlines an *open, light-weight, synchronous API dedicated for Consumer Payments*. Due to the use of *account-specific authorization keys* in the associated “Wallet” as well as *encrypted user authorizations*, the API does not expose (or depend on) personal information to external parties, effectively removing “consents” from the plot. This arrangement also enables a (with respect to the user), *fully end-to-end secured payment authorization protocol*.

Since APIs for external consumption come with considerable development and maintenance costs this represents a major hurdle to adoption. By rather reusing Open Banking APIs in a novel way integration costs can be kept reasonable as well as spread over multiple banks having the same flavor of Open Banking API.

Note: The API is *unchanged*, the only update needed is recognizing that the caller is a *locally installed and trusted service*. User login is though required during virtual card enrollment. This is preferably accomplished through the bank’s regular on-line login.

Although the presentation shows a mobile phone based system, the proposed dual mode would also support legacy schemes like EMV

Using a Payment Gateway

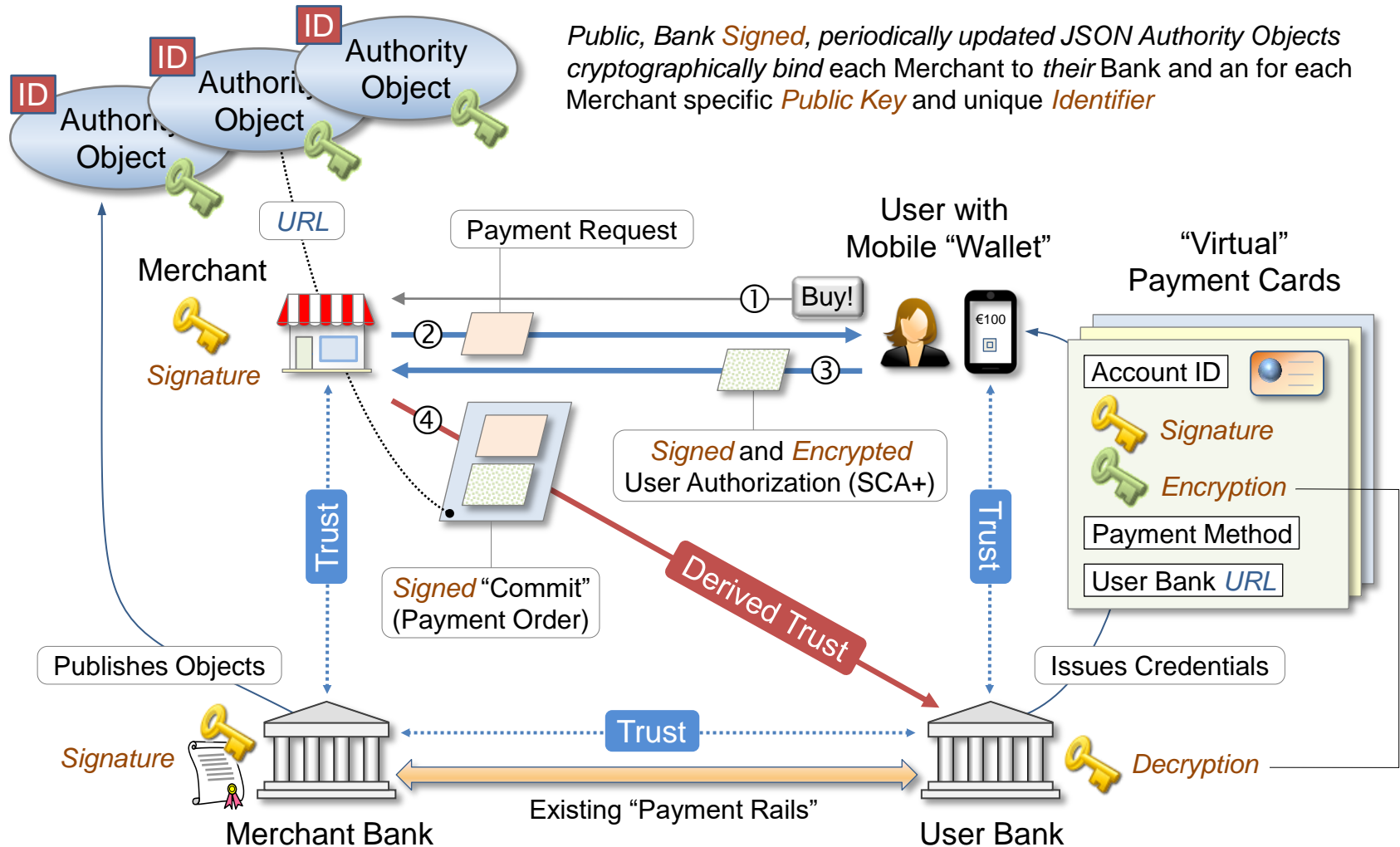


The Payment Gateway setup is quite similar to traditional card processing solutions. There are though some notable differences:

- The User Bank URL supplied by the "Wallet" eliminates the database for finding out associated User (Issuer) Bank.
- Since no personal information (except User Bank URL) is revealed to the Payment Gateway, its sole mission is vouching for the authenticity of the Merchant. Only User Bank can "decipher" a user authorization and match it with the supplied Payment Request.

Note: The Payment Gateway may either counter-sign messages or authenticate through mutual TLS to the User Bank.

Using Saturn



Using [Saturn](#) the functionality of a Payment Gateway is "emulated" by the Merchant's Bank. However, the Merchant's Bank only *indirectly* vouches for the authenticity of the Merchant.

Note: This drawing is slightly simplified, succeeding steps as well as [Service Discovery](#) are not shown here.

Note: The "Wallet" is *identical* for Saturn and Payment Gateway setups.