Overview

In real life we may visit a shop or service counter anonymously – or enjoy being treated like a well-known customer by presenting a loyalty card; we may make our picks and conduct our business and then decide to pay whatever way the site we are in supports. The shop will most likely have rented a payment terminal and with it some options to allow the customer to pay. A few payment networks, hundreds of issuers and thousands of payment products may be supported this way. Security is implemented in smart cards on plastic carriers, the user might be asked to sign a piece of paper or key in a PIN at the payment terminal for secure authentication.

In the web we only find elements of this – and they are usually disconnected in some way. Some of these 'disconnections' pose security threats: submitting credit card numbers over the network, storing credentials connected with an account somewhere at a web site, creating yet another user account with passwords that can be stolen and be used to implicitly trigger payments based on credentials stored at the service. Often, convenience is limited and many purchases are not made because the payment credentials or personal information is requested which the user doesn't want to provide easily. Online payment services can take away some of the pain and professionalized the process, but we are far from transferring our wallets into the digital space – entirely and with proper security. Little is in place to pass along coupons or loyalty cards when making web payments and why shouldn't we take into account the transformation of proximity payments to fully digital transactions?

In modern smart phones we find everything to let the physical and the virtual world meet to provide a joint metaphor – a converged wallet. Over the last years in which almost all big mobile operators have started to introduce SIM- and NFC-based payment or 'wallet' services, work at T-Labs has produced a user-centric wallet concept which makes use of the very same assets, but is deeply rooted in web- and identity technology. With the help of a set of interfaces, protocols and data structures a wallet framework could allow future issuers to create virtual credit cards which work in the shop as well as in the Internet, using specific security technology for their respective purpose. The same way coupons can be stored read out via NFC or optical scanners, but also be used in a web-based purchase. Loyalty cards are no different from login credential cards in this wallet. Almost all loyalty cards allow access to the account via Internet. The wallet abstraction can store the credentials like a password store – or even better – be a first step into login with cryptographic keys. All login credentials could – vice versa – be handled like virtual loyalty, or customer cards.

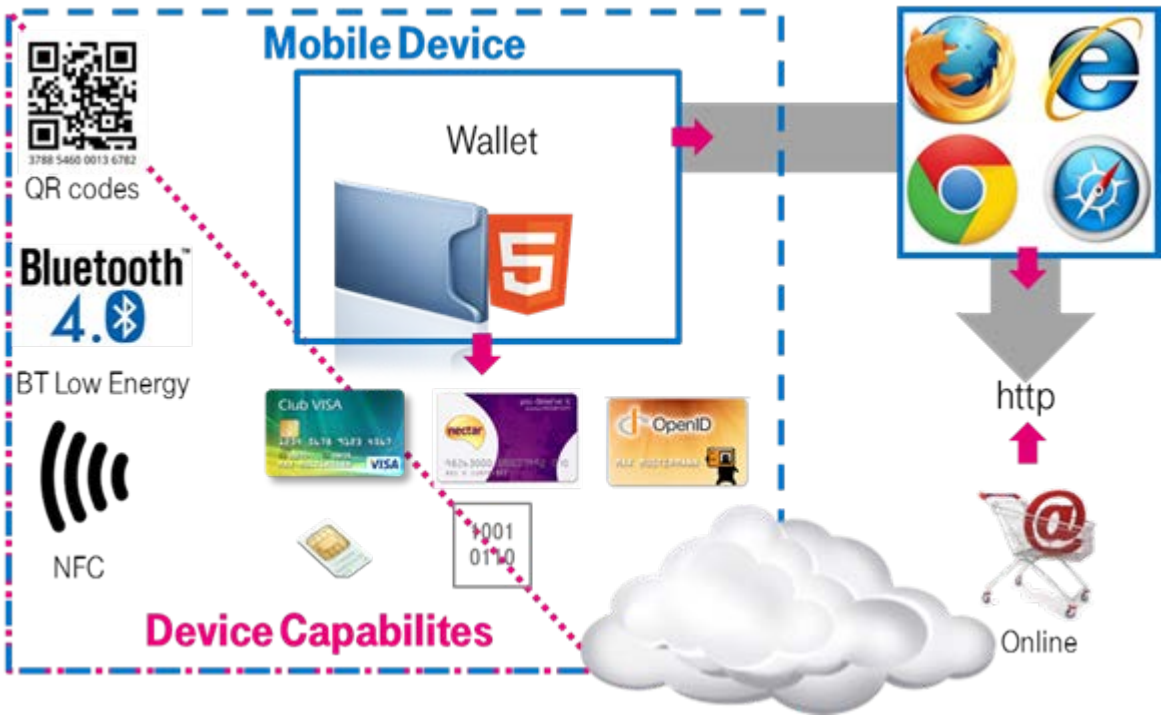
Tickets and keys can be handled likewise, but all of them will benefit from the transfer to a convergent wallet. Improved fraud handling, emergency functions like backup/ restore or eased management of keys for enterprises, more attractive marketing tools and overall eased handling for the end user are just some of them. Openness with respect to how security is implemented (secure elements might not always be adequate) and by what means communication takes place turns the concept into a paradigm of significant extent. Such wallets can run on all kinds of digital devices or in the cloud. Naturally, we couldn't implement all of this yet, but there are plenty of functional samples available, following this approach. Some embrace existing technologies, others have already created new protocol drafts, but many interfaces and APIs are still undefined or unassigned. Many concessions had to be made, work-arounds to be created for different platforms, operating systems and specific technology. We found plenty of useful functionality in web technology to create our first implementations, but so much is still open...

In the web we only find elements of this – and they are usually disconnected in some way. I might have pre-registered a credit card (inconvenient) and the shop will have to keep the credentials safe – while my login credentials to the shop open a way to my payment means – so I'd better keep it super safe too. Alternatively I might decide to provide my payment credentials the moment I use them – which has to be done with every new service – and expecting the shop to handle the data safely and delete it after our transaction is over. In all these cases, credit card numbers (and usually a few other bits of information from the plastic card which are not really secret) are used to authorize payments with the user's account. There are plenty of attack scenarios possible and transparency to the end-user is minimal.

Payments on the Web today present a set of challenges for all the actors of the ecosystem, from application developers (merchant), to payment systems to end-users (buyers). To reduce the burden on developers, and to give users greater freedom of choice in how they pay, we need to **decouple payment requests and payment providers**, with a trusted intermediary functioning as a virtual **wallet**. Payment requests would be passed to the wallet, and the user is allowed to choose the means for payment from amongst those available. For a level playing field, the wallet needs to be independent of the payment solution providers. This implies the requirement for users to be able to install and uninstall payment solutions after a device has been shipped. Likewise, the choice of payment solutions shouldn't be tied to the web browser.

Relevance

Assuming that a wallet application can be programmed in HTML5 for operation on a device, it should support the device's specific capabilities (in this concrete case, e.g. utilizing PhoneGap to ease cross-platform development). The following picture describes the constellation between relevant components and domains in an abstract manner.



It is conceivable that implementations for individual items within a wallet will either be provided in HTML5, possibly running in sandboxes or rather be configurations of relevant standards (like OAuth, OpenID, etc...) Online functionality will be initiated by a user's action in the web browser. In the above example a web site requires authentication, payment or is open to receive profile, coupon or similar information and indicates so to the user's web browser. If the user choses so, the wallet gets started (or comes to the front, if already running) and presents relevant items for the user to pick. After the user's selection, only those items' content gets transferred in the appropriate ways to the requesting web site.

Naturally, protocols and technical interfaces employed in such a complex scenario may differ widely. However, the overall interaction paradigm between the user and the involved parties and their technical representations can be unified to a great extent. Web browsers implementing interfaces to all kinds of password stores and wallets are just as valuable as are interfaces on web sites that ask for wallets, indicating item types and protocols supported. There are various ways how actual payments and other transactions are implemented. Even if there is heterogeneity in technology, there should be a way to unify user interaction and interaction between the components involved.