INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

ePaga - Sistema de Pagamento Electrónico

João António Frade Macedo de Almeida

Dissertação para obtenção do Grau de Mestre em

Mestrado em Engenharia de Redes de Comunicações

Júri

Presidente: Professor Rui Jorge Morais Tomáz Valadas

Orientador: Professor Carlos Nuno da Cruz Ribeiro

Co-Orientador: Professor Paulo Jorge Pires Ferreira

Vogais: Professor Nuno Filipe Valentim Roma

Junho de 2012

Agradecimentos

Gostaria de expressar a minha gratidão pelas contribuições cruciais dos Professores Paulo Ferreira e Carlos Ribeiro, sem as quais não teria sido possível concretizar este projecto.

Não posso também deixar de agradecer os conselhos dos meus colegas do Instituto Superior Técnico, cuja disponibilidade e espirito de interajuda tiveram um valor inestimável ao longo do meu percurso académico.

Por fim, quero agradecer à minha família e amigos por me motivarem e apoiarem de forma incondicional.

Resumo

A possibilidade de utilizar dispositivos móveis para efectuar pagamentos tem vindo a prometer maior rapidez, conveniência e ubiquidade, em relação aos métodos actuais de pagamento. Por estas razões, o mercado dos pagamentos móveis evidencia um potencial elevado. No entanto, a falta de interoperabilidade entre sistemas de pagamento, em conjunto com a escassez de tecnologias apropriadas, tem atrasado o progresso na área dos pagamentos móveis.

O objectivo deste trabalho é desenvolver um sistema de suporte a pagamentos electrónicos, focado nos pagamentos móveis, que permite tirar partido de vários protocolos de pagamento que convivam simultaneamente no mesmo dispositivo. A aplicação resultante deste suporte a múltiplos protocolos atinge assim uma maior universalidade, eficiência e interoperabilidade, quando comparada com um protocolo isolado. O sistema proposto baseia-se na tecnologia NFC (*Near Field Communication*). Este documento descreve ainda o protótipo implementado que demonstra o funcionamento do sistema, assim como a sua avaliação.

Palavras-chave: computação móvel, pagamentos electrónicos, pagamentos móveis, NFC.

Abstract

The possibility of using mobile devices for payments promises greater speed, convenience and ubiquity when compared with current payment methods. Hence the large potential demonstrated by the mobile payments market. However, the lack of interoperability between payment systems, along with the absence of suitable technologies, has delayed the progress in the field of mobile payments.

This project's goal is to develop a support system for electronic payments, with a special focus on mobile payments, which takes advantage of multiple payment protocols installed on the same device. The application derived by this multiprotocol support achieves higher universality, efficiency and interoperability, when compared to a single protocol application. The proposed system is based on NFC (Near Field Communication) technology. This document also describes the prototype developed with the purpose of demonstrating the system's behavior, as well as its evaluation.

Keywords: mobile computing, electronic payments, mobile payments, NFC.

Índice

Índice de Figuras

Lista de Acrónimos

AKE	*Authenticated Key Exchange*
APDU	*Application Protocol Data Units*
ATM	*Automated Teller Machine*
BIC	*Business Identifier Code*
C2C	*Customer to Costumer*
CA	*Certification Authority*
CKLA	*Confidential Key Loading Authority*
CPP	*Customer's Payment Processor*
CREF	*C-language Java Card RE*
CSM	*Clearing and Settlement Mechanisms*
EPC	*European Payments Council*
ETSI	*European Telecommunications Standards Institute*
GP	*GlobalPlatform*
GPRS	*General Packet Radio Services*
GSM	*Global System for Mobile Communications*
GSMA	*GSM Association*
HSPA	*High Speed Packet Access*
HTTP	*Hypertext Transfer Protocol*
IBAN	*International Bank Account Number*
IC	*Integrated Circuit*
IP	*Internet Protocol*
IrDA	*Infrared Data Association*
ISO	*International Organization for Standardization*
JCDWE	*Java Card platform Workstation Development*
LLCP	*Logical Link Control Protocol*
LTE	*Long Term Evolution*
MAC	*Media Access Control*
MMS	*Multimedia Messaging Service*
MPP	*Merchant's Payment Processor*
NFC	*Near Field Communication*
OBEX	*OBject EXchange*
OSI	*Open Systems Interconnection*
OTA	*Over The Air*
P2P	*Peer-to-Peer*
PG	*Payment Gateway*

PIN	*Personal Identification Number*
PKCS	*Public-Key Cryptography Standards*
PKI	*Public Key Infrastructure*
POS	*Point Of Sale*
RFID	*Radio-frequency identification*
RSA	Rivest, Shamir e Adleman
SCT	*SEPA Credit Transfer*
SDD	*SEPA Direct Debit*
SDK	*Software Development Kit*
SEMOPS	*SEcure MObile Payment Service*
SEPA	*Single Euro Payments Area*
SHA	*Secure Hash Algorithm*
SIM	*Subscriber Identity Module*
SMS	*Short Message Service*
SSD	*Supplementary Security Domain*
SWP	*Single Wire Protocol*
TCP	*Transmission Control Protocol*
TLS	*Transport Layer Security*
TMS	*Trusted Service Manager*
UMTS	*Universal Mobile Telecommunications System*
URL	*Uniform Resource Locator*
USSD	*Unstructured Supplementary Service Data*
WAP	*Wireless Application Protocol*
XML	*eXtensible Markup Language*

1 Introdução

1.1 Introdução aos Pagamentos Móveis

Mark Weiser ambicionou um mundo em que uma pessoa, independentemente do lugar em que se encontrasse, estaria rodeada de uma infinidade de dispositivos móveis [1]. Estes dispositivos interagiriam entre si de forma a fornecerem, de forma transparente, um número elevado de serviços que melhorassem a experiência do utilizador. No entanto, como é que estes dispositivos disponibilizam estes serviços se não existir uma forma de pagarem por eles? A resposta a esta necessidade conduz ao conceito de um sistema de pagamentos móveis.

Um pagamento móvel é um pagamento electrónico em que pelo menos um dos intervenientes na transacção é representado por um dispositivo electrónico móvel. Um sistema de pagamentos móveis é um sistema de pagamentos electrónicos que suporta pagamentos móveis.

Partindo do pressuposto de que é necessário o referido sistema de pagamentos móveis, é preciso analisar o dispositivo móvel que suporta o sistema. Este tipo de dispositivos, mais concretamente o telemóvel, tem evoluído significativamente nos últimos anos. Um telemóvel é cada vez menos apenas um telefone móvel. Actualmente é usado como agenda electrónica, para tirar fotografias, ler emails, jogar, explorar a Internet, entre muitas outras utilizações. De facto, o telemóvel assemelha-se mais a uma carteira ou mala. No entanto, no interior da carteira ainda reside um tipo de objectos que o telemóvel não substituiu. Ainda continuamos a contar os trocos que vamos precisar para pagar uma portagem, ou à procura do cartão certo no meio de uma infinidade de cartões de crédito e fidelização. De facto, das funcionalidades desempenhadas por objectos que transportamos diariamente, a capacidade de executar pagamentos é uma das poucas que ainda não foram assimiladas pelo versátil dispositivo móvel.

Por outro lado, o telemóvel é o candidato perfeito para desempenhar esta função por duas razões. Em primeiro lugar é o dispositivo electrónico mais pessoal e popular [2] no planeta. Em segundo lugar possui as capacidades que permitem o suporte desta funcionalidade, nomeadamente tecnologias de comunicação de curto e longo alcance, assim como um elemento seguro que permite, por exemplo, guardar dados confidenciais.

Um sistema de pagamentos móveis é mais rápido do que dinheiro, por não se ter de procurar e contar moedas e notas, é mais prático porque não está limitado pela quantia que se transporta no momento, é mais higiénico e sofre consequências menores em caso de roubo. Em relação ao cartão de crédito, possibilita a escolha de vários fornecedores sem ocupar mais espaço ou demorar o mesmo tempo na escolha do mesmo. Possibilita também pagamentos directos e rápidos entre clientes particulares, situação que não é possível nos cartões de crédito.

1.2 Desafios dos Pagamentos Móveis

As claras vantagens que os pagamentos móveis proporcionam deixam antever um futuro risonho para esta área, contudo a concepção de um sistema deste tipo não está isenta de dificuldades. Os principais destes problemas são descritos de seguida.

Apesar da evolução que os dispositivos móveis têm sofrido, continuam a dispor de recursos limitados, como capacidade computacional, qualidade de serviço na comunicação e capacidade da bateria. Os sistemas desenvolvidos devem ter estas limitações em conta.

Um sistema de pagamentos móveis não oferece uma funcionalidade completamente nova, apenas uma nova forma de a desempenhar, visto que as pessoas já conseguem efectuar pagamentos sem o sistema. Logo, este deve apresentar claras vantagens em relação as alternativas existentes para que os clientes encarem o serviço como uma mais-valia.

Para além de competir com os métodos tradicionais, um sistema de pagamentos móveis tem ainda de lidar com a inexistência de *standards* nesta área [3]. Esta deficiência contribui para a falta de interoperabilidade que se verifica nos sistemas actuais.

Durante a última década têm sido desenvolvidas diversas iniciativas com o objectivo de explorar o potencial dos pagamentos electrónicos em dispositivos móveis, no entanto, algumas não chegaram a ser implementadas, outras não se conseguiram impor no mercado, restando uma pequena fracção de projectos que subsistem com relativo sucesso. Entre o variado leque de razões que conduzem ao fracasso de um sistema incluem-se as descritas de seguida.

Uma elevada percentagem de projectos de pagamentos móveis são desenvolvidos por um banco ou uma operadora de telecomunicações. Uma percentagem mais reduzida tem origem num grupo limitado de bancos ou de operadoras. Em casos relativamente raros os sistemas são criados por um conjunto reduzido de entidades de ambos os tipos. Estes sistemas competem entre si e dividem o mercado em fragmentos não interoperáveis, reduzindo assim a probabilidade de sucesso dos mesmos.

A falta de universalidade de um sistema constitui outro factor que reduz o público-alvo do mesmo. Um número significativo de projectos centra-se num conjunto limitado de situações de pagamento. Estes sistemas apresentam restrições como suportar apenas pagamentos de quantias reduzidas, ou suportar somente a compra de conteúdos disponibilizados pela operadora.

Outros sistemas sofrem de desvantagens provenientes das tecnologias escolhidas. Em certos casos as tecnologias adoptadas reduzem a segurança do sistema, enquanto outras o tornam pouco prático. Existem ainda sistemas que requerem tecnologias com um custo demasiado elevado, geralmente apenas disponíveis em dispositivos de gama alta.

1.3 Motivação

Os desafios identificados na secção anterior conduzem ao problema que esta tese pretende abordar. Os numerosos sistemas de pagamento existentes, cada um com sucesso e funcionalidades limitadas, atacam o mercado de forma independente. Este cenário introduz no ecossistema dos

pagamentos móveis um ambiente demasiado fragmentado. Esta divisão dispersa os actores e atrasa a conquista do mercado por parte dos pagamentos móveis. Os clientes têm de instalar, gerir, e aprender a usar várias aplicações de pagamento. Os vendedores têm de aderir e suportar vários sistemas de pagamentos móveis, assim como possuir o consequente equipamento extra.

Como tal, é necessária uma solução unificante, que canalize os vários esforços empregues pelos vários actores envolvidos, para que o impacto das suas iniciativas seja maximizado. Sem uma solução deste género, o mercado dos pagamentos móveis arrisca-se a continuar limitado a nichos ou a certas zonas do globo.

1.4 Objectivos

Os objectivos que contribuem para a resolução do problema identificado são os seguintes:

- Facilitar o desenvolvimento e maximizar o sucesso dos sistemas de pagamentos móveis.
- Facilitar e unificar o acesso por parte dos utilizadores aos pagamentos móveis.

Um sistema que se proponha a atingir os objectivos enumerados implica os seguintes requisitos:

- Simplicidade: O sistema deve ser fácil de usar pelos clientes e vendedores, e a sua utilização deve estar dependente de uma aprendizagem rápida.
- Universalidade: O sistema deve incluir o maior número de utilizações possível nos ambientes mais diversos, incluindo diferentes tipos de intervenientes e pagamentos de diversas quantias. Estas utilizações incluem, por exemplo, pagamentos P2P e pagamentos *offline*.
- Interoperabilidade: O sistema não deve estar limitado por uma marca de dispositivos, por um banco ou por uma operadora de telecomunicações. Também deve ser facilitada a interacção com outros sistemas de pagamentos electrónicos, assim como a integração com sistemas de pagamento tradicionais.
- Consistência: Apesar de funcionar num ambiente instável e heterogéneo, o sistema deve apresentar uma interface consistente ao utilizador e aos sistemas que suporta.

Um sistema de pagamento inclui requisitos adicionais. No entanto, não cabe à aplicação proposta garantir que estes sejam respeitados por todos os protocolos. Requer-se apenas que não dificulte o seu cumprimento por parte de cada protocolo. Estes requisitos são descritos de seguida.

- Privacidade: O conteúdo das transacções efectuadas através do sistema não deve poder ser consultado publicamente.
- Anonimato: Uma transacção não deve poder ser associada directamente à identidade do cliente que a executou.
- Rastreabilidade: A garantia anterior deve poder ser quebrada em situações de investigação criminal. Este requisito é necessário de modo a evitar que o sistema seja usado com objectivos ilícitos como lavagem de dinheiro.
- Não-repudiação: O sistema deve conseguir provar inequivocamente que um cliente efectuou uma transacção.

- Custo: O custo associado a cada pagamento deve ser competitivo em relação aos métodos de pagamento existentes, para qualquer quantia envolvida. O custo de implementação do sistema também deve ser minimizado.
- Rapidez: As transacções efectuadas pelo sistema devem pelo menos tão rápidas como os métodos tradicionais de pagamento. Contudo, a comparação deve ser feita caso a caso, visto que o sistema compete com alternativas distintas em cada tipo de pagamento.

1.5 Soluções Existentes

Até ao momento, as iniciativas que podem mitigar o problema identificado seguem duas abordagens distintas:

- Um sistema de pagamento único e universal: Esta é a abordagem mais comum até ao momento. Um sistema de pagamento que, para além de prever e ser eficiente em todas situações, também fosse aceite por todos os actores, poderia eliminar o problema. No entanto, as dificuldades de atingir a universalidade e aprovação desejadas reduzem a probabilidade de sucesso deste tipo de solução.
- Standardização: Se existisse um standard que definisse uma base para a implementação de sistemas de pagamentos móveis, o problema seria parcialmente resolvido. A previsibilidade introduzida levaria a uma maior interoperabilidade entre sistemas de pagamentos, que aumentaria a probabilidade de sucesso de cada um deles. Na prática, as iniciativas desenvolvidas até ao momento, como a SEPA e a GlobalPlatform, não são suficientemente abrangentes e específicas. A SEPA coíbe-se de especificar interacções próximas do cliente, concentrando-se especialmente na interacção entre componentes centrais como bancos. A GlobalPlatform foca-se na gestão do ciclo de vida das aplicações. Ambas as iniciativas referidas serão descritas em maior detalhe numa parte posterior deste documento.

1.6 Descrição da Solução

A principal diferença entre o ambiente actual do inicial, com que se deparam os investigadores pioneiros no desenvolvimento de sistemas de pagamentos móveis, centra-se na constante evolução da qualidade dispositivos e redes móveis, que se traduz, por exemplo, no aparecimento de novas tecnologias como o NFC.

De modo a responder à elevada quantidade de projectos existentes no mercado, a solução proposta neste documento oferece uma infra-estrutura que se adapta a este ambiente heterogéneo. Esta solução tem como objectivo definir a arquitectura de uma aplicação que suporte vários sistemas de pagamentos electrónicos, com enfâse nos móveis. A aplicação inclui um *middleware* que, no momento de efectuar um pagamento, escolhe o sistema mais apropriado, consoante os sistemas suportados pelo vendedor e os requisitos do cliente.

Ao contrário dos sistemas de pagamento existentes, que propõem um protocolo de pagamento e competem com os restantes, o sistema ePaga pretende suportar vários protocolos, combinando as suas vantagens. Deste modo evita fragmentar o mercado, aumentando o número de clientes

abrangidos por cada um dos protocolos. A solução proposta também não compete com as abordagens que foram referidas. A sua relação com o sistema ePaga é de complementaridade. A solução proposta suporta protocolos que sejam compatíveis com a SEPA. Em relação a iniciativas como a GlobalPlatform, facilitam a gestão do ciclo de vida da aplicação proposta.

1.7 Contribuições

Esta tese analisa o ecossistema dos pagamentos móveis e categoriza os sistemas de pagamentos móveis existentes. A partir desta análise, a tese propõe uma arquitectura para um sistema de suporte a pagamentos electrónicos. O sistema permite e facilita o desenvolvimento de protocolos de pagamento que partilham a mesma aplicação. Esta tese demonstra ainda o funcionamento da arquitectura através de uma prova de conceito.

1.8 Estrutura do Documento

Este documento é composto por 6 capítulos. No capítulo seguinte serão caracterizados e exemplificados os sistemas existentes, bem como as tecnologias envolvidas. No capítulo 3 será apresentada a arquitectura da solução proposta. No capítulo seguinte é descrita a implementação de uma prova de conceito, que demonstra a arquitectura do capítulo anterior. De seguida, no capítulo 5, a solução desenvolvida será avaliada. Na última secção serão enumeradas as conclusões do trabalho desenvolvido.

2 Trabalho Relacionado

Nesta secção do documento são abordados os temas relacionados com a área dos pagamentos móveis. Na parte inicial da secção são descritos os actores que devem ser tidos em conta no desenvolvimento de um sistema de pagamentos móveis. A segunda subsecção aborda algumas iniciativas que podem contribuir para a interoperabilidade dos sistemas de pagamentos móveis. A subsecção posterior refere de forma breve as tecnologias mais utilizadas neste tipo de sistemas. No final desta secção são descritos e comparados alguns sistemas de pagamentos móveis que, pelas suas características, representam as diferentes formas de conceber um sistema de pagamentos móveis.

2.1 Actores

Uma das razões que dificultam a criação de *standards* e reduzem o sucesso das iniciativas na área dos pagamentos móveis é a diversidade de entidades envolvidas, cada uma com perspectivas e objectivos distintos. Passa-se a enumerar as mais importantes [1][4]: clientes particulares, vendedores, bancos, operadoras de telecomunicações, fabricantes de dispositivos móveis e entidades governamentais ou reguladoras.

Clientes Particulares [5]: Para que um serviço de pagamentos móveis seja adoptado pelos utilizadores precisa de se diferenciar dos métodos de pagamento actuais, tais como dinheiro, cartões de crédito ou cheques. Os principais factores que influenciam a adesão de clientes a um sistema de pagamentos móveis são a facilidade de uso e a percepção de segurança por parte do utilizador, nomeadamente privacidade e confiança. Uma das falhas apontadas a alguns sistemas reside na relutância do utilizador em entregar dados confidenciais a empresas de pagamentos móveis em que este não confia, em vez de os entregar a uma entidade que já conhece como o seu banco ou operadora de telecomunicações. Outras das características importantes para os utilizadores são o custo do serviço, disponibilidade e ubiquidade, isto é, a possibilidade de utilizar o sistema a qualquer hora e qualquer lugar, interoperabilidade entre operadoras, bancos e dispositivos, anonimato e possibilidade de realizar pagamentos P2P (*Peer-to-Peer*), ou seja, entre clientes. Desejam ainda que o anonimato seja total e inquebrável, o que contraria o requisito da rastreabilidade.

Vendedores: Para além de partilharem com os clientes preocupações como custo e segurança, os pontos que têm maior relevância são a rapidez da transacção e a facilidade de integração com os sistemas de pagamento existentes. Esta facilidade de integração traduz-se, por exemplo, na necessidade dos equipamentos terminais serem simultaneamente compatíveis com a nova e com as antigas formas de pagamento. Os vendedores desejam também que o serviço admita consultar o estado das transacções em tempo real, e que o sistema seja versátil o suficiente para facilitar personalização (por exemplo a adição de serviços de fidelização).

Bancos: Para os bancos, os pagamentos móveis representam uma oportunidade para oferecer um novo serviço, atraindo novos clientes, aumentando a lealdade dos mesmos e maximizando o lucro por cliente. Os pagamentos electrónicos são preferíveis em relação a transacções em dinheiro, nas

quais os bancos não obtêm lucro. São também preferíveis em relação aos métodos tradicionais por serem mais baratos. Os bancos pretendem ter controlo sobre as aplicações de pagamento e desejam que o sistema seja independente das operadoras de telecomunicações. Este desejo faz com que os projectos de pagamentos móveis com origem na área financeira se baseiem por vezes em telemóveis *dual-SIM,* que suportam dois *smartcards*. Como o cartão SIM (*Subscriber Identity Module*) está dependente da operadora, a aplicação do banco usa o segundo cartão, controlado por si. Este requisito restringe estes sistemas a um número limitado de dispositivos que suportam esta tecnologia.

Operadoras de Telecomunicações: As operadoras, à semelhança dos bancos, também encaram os pagamentos móveis como uma possibilidade de oferecer um novo serviço, com as vantagens referidas anteriormente. Como proprietárias da rede usada pelos clientes, as operadoras retiram receitas da sua utilização, o que torna qualquer serviço que necessite de comunicação através da sua rede uma fonte de rendimento extra. Tal como os bancos, também as operadoras ambicionam controlar as aplicações de pagamento, assim como independência do sistema em relação aos bancos. As operadoras já oferecem serviços de pagamento por telemóvel há algum tempo. Estes são geralmente centrados no pagamento de conteúdos distribuídos pela operadora e utilizados no dispositivo. Em comparação com os bancos, as operadoras desfrutam de uma regulação menos apertada, visto que os bancos são monitorizados por bancos nacionais e regionais, por exemplo no caso europeu pelo Banco Central Europeu.

Fabricantes de dispositivos móveis: Os fabricantes influenciam o desenvolvimento de sistemas de pagamentos móveis através da inclusão de novas tecnologias nos dispositivos. Os fabricantes mantêm-se atentos, de forma a preverem as tecnologias que serão necessárias no futuro. Os atributos vistos como favoráveis, num serviço de pagamentos móveis, são a escolha de uma tecnologia pouco dispendiosa e rápida introdução no mercado.

Entidades Governamentais ou Reguladoras: Estas instituições têm o papel de desenvolver legislação favorável, assim como de promover o desenvolvimento de *standards*, que facilitam a interoperabilidade dos sistemas. Em alguns casos as entidades governamentais podem também impulsionar a proliferação de sistemas de pagamentos móveis, através de iniciativas como a criação de uma PKI (*Public Key Infrastructure*) e atribuição de chaves e certificados aos cidadãos. O FINE-ID Finlandês representa um exemplo deste tipo de iniciativas [6][7]. As entidades governamentais prezam o requisito da rastreabilidade que permite, no âmbito de uma investigação criminal, aceder à informação relativa às transacções que um individuo efectuou.

2.2 Interoperabilidade

Apesar da actual falta de *standards* na área dos pagamentos móveis, existem algumas iniciativas que devem ser consideradas durante o desenvolvimento de sistemas de pagamentos móveis.

2.2.1 ISO 7816

Como já foi referido, a existência de um elemento seguro nos telemóveis é uma das suas principais vantagens num contexto de pagamentos móveis. O elemento seguro oferece ao dispositivo em que

está inserido processamento e armazenamento seguros. Este elemento, geralmente representado por um cartão SIM, é essencialmente um *smartcard*.

O *standard* ISO 7816 especifica as características que um *smartcard* deve ter para poder ser utilizado nos mais diversos sistemas. A parte mais relevante deste *standard* para um sistema de pagamentos móveis centra-se na especificação das mensagens aceites pelo *smartcard*. O *standard* define o tipo de mensagem aceite denominado APDU (*Application Protocol Data Units*), que segue uma lógica de comando e resposta.

2.2.2 SEPA

O EPC (*European Payments Council*) é uma instituição europeia cujo objectivo é a criação de uma área em que os pagamentos sejam executados de forma independente em relação a fronteiras nacionais. A esta área atribuiu-se o nome de SEPA (*Single Euro Payments Area*). Neste contexto o EPC definiu procedimentos para executar transferências a crédito SCT (*SEPA Credit Transfer*) [8] e débitos directos SDD (*SEPA Direct Debit*) [9]. Uma SCT é definida como uma transferência de fundos entre as contas de um *Originator* e um *Beneficiary*. Uma SDD é descrita em [10] pelo conceito de *"eu solicito determinada quantia a alguém, com a sua aprovação prévia, e essa quantia é creditada na minha conta"*. Neste processo um devedor assina um mandato que autoriza o credor a cobrar um pagamento. Este pagamento pode ser único ou recorrente.

Estes *standards* não definem uma infra-estrutura específica para pagamentos móveis. De facto, a SCT não define sequer uma infra-estrutura específica para pagamentos electrónicos, isto é, não prevê utilizações como pagamentos *online* a partir do computador pessoal. Este tipo de funcionalidades não quebra o *standard*, mas tem de ser implementado pelos bancos.

Estes *standards* têm, no entanto, características que facilitam a sua adaptação a sistemas de pagamentos electrónicos ou móveis. As mensagens trocadas não precisam de ter um suporte físico, e são baseadas em XML (*eXtensible Markup Language*) [11] e ISO 20022. O ISO 20022 define um conjunto de mensagens dedicadas à indústria financeira.

No caso do SDD esta adaptação está ainda mais facilitada. Apesar de também não prever um ambiente móvel, o *standard* endereçou especificamente os pagamentos electrónicos, quando definiu o conceito de *e-Mandate* [12]. Antes da definição do *e-Mandate* existia uma fase do SDD que era obrigatoriamente processada em papel. O mandato, que corresponde à autorização do débito por parte do devedor, impedia que o SDD pudesse ser totalmente processado de forma electrónica. Assim, o EPC definiu um artefacto denominado *e-Mandate*, que corresponde a um mandato criado de forma electrónica. Na definição de *e-Mandate*, refere-se que a comunicação em que o devedor está envolvido é baseada em HTTP (*Hypertext Transfer Protocol*). Este facto, em conjunto com a consideração, por parte do EPC, de que o trafego em parte da rede é transmitido através da Internet, inserem este *standard* numa típica aplicação POS (*Point Of Sale*) virtual. O *standard* refere inclusivamente a insersão do sistema no serviço de *Home Banking* dos bancos. A utilização da Internet leva também a uma maior preocupação com segurança do que numa rede privada. O EPC

refere a utilização de conexões TLS (*Transport Layer Security*) como suporte à comunicação HTTP, assim como certificados digitais na autenticação de algumas das entidades envolvidas.

De seguida são descritas as arquitecturas definidas pelo EPC para a SCT e o SDD. A arquitectura da SCT é constituída pelos seguintes componentes: cliente, vendedor, banco do cliente, banco do vendedor e CSMs (*Clearing and Settlement Mechanisms*).

- Cliente: representa a entidade que inicia a transacção. O valor do pagamento é descontado da sua conta bancária. Como na SCT descrição da relação entre o cliente e o seu banco é pouco específica, não é possível indicar que tipo de dispositivos ou tecnologias são usados. No entanto, numa possível adaptação desta arquitectura a um sistema de pagamentos móveis, o cliente seria representado por um dispositivo móvel. A conta do cliente é identificada pelo seu IBAN (*International Bank Account Number*).
- Vendedor: entidade que recebe o pagamento. A sua conta é identificada pelo IBAN correspondente.
- Banco do Cliente: instituição bancária aderente à SEPA na qual o cliente tem uma conta. É identificado pelo seu BIC (*Business Identifier Code*).
- Banco do Vendedor: instituição bancária aderente à SEPA na qual o vendedor tem uma conta. É identificado pelo seu BIC. Pode ser o mesmo banco do cliente.
- CSMs ou Mecanismos de Compensação e Liquidação: conjunto que entidades que permitem aos bancos comunicarem e efectuarem transferências entre si.

Na Figura 1 está representada uma possível adaptação da arquitectura descrita a um sistema de pagamentos móveis. Aproveita-se também a figura para introduzir os conceitos associados a um sistema de pagamentos móveis.

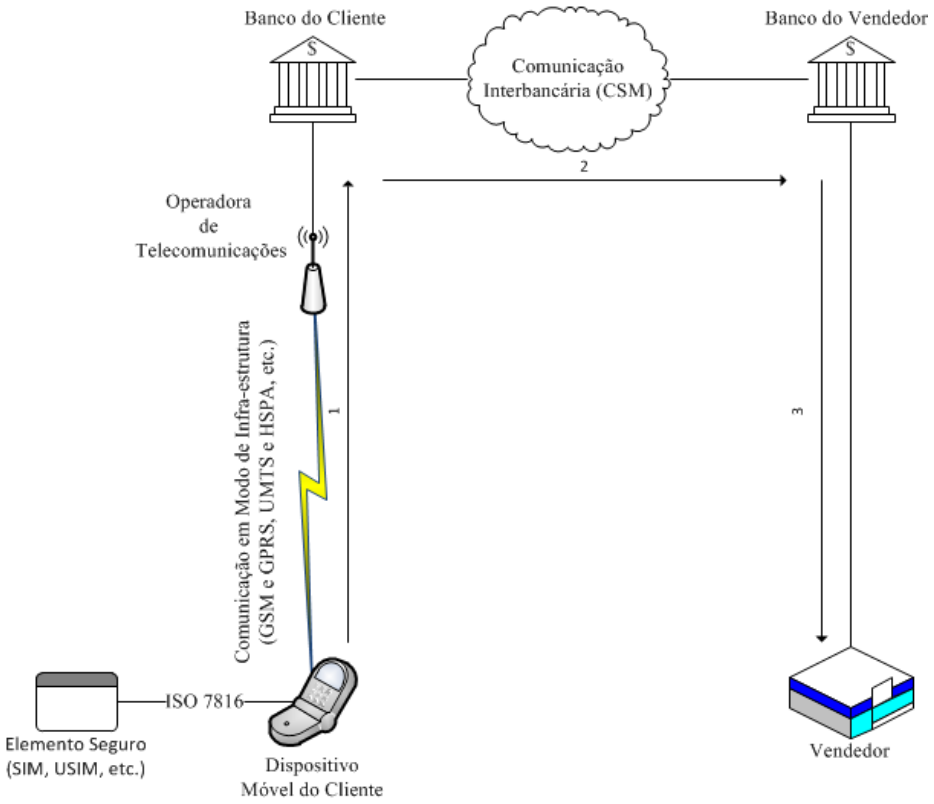


Figura 1. Adaptação da arquitectura da SCT de modo a representar os conceitos envolvidos num sistema de pagamentos móveis. Em relação à representação original, o componente cliente passou a ser representado por um dispositivo móvel. Foram também incluídos o elemento seguro e a rede de acesso do dispositivo. O conceito de Comunicação em Modo de Infra-estrutura será abordado na secção 2.4 deste documento.

Os passos para executar uma transacção SCT são os seguintes:

- Passo 1. O cliente comunica ao seu banco que deseja efectuar um pagamento ao vendedor. Na mensagem inclui, entre outros dados, o IBAN do vendedor e o BIC do banco do vendedor. A forma como o cliente comunica esta mensagem ao banco não está definida, apenas a informação que esta deve conter.
- Passo 2. O banco do cliente verifica a validade do pedido do cliente. A forma como o banco determina a veracidade dos dados que o cliente introduziu não está definida. O *standard* apenas refere que o banco verifica se a mensagem está completa. De seguida, o banco desconta o valor do pagamento na conta do cliente. No caso de um cancelamento posterior da transacção o banco desfaz esta operação. O banco envia então um pedido de transferência ao banco do vendedor, através de um CSM ao qual ambos os bancos pertencem.
- Passo 3. O banco do vendedor actualiza o valor da conta do vendedor. De seguida, disponibiliza a informação do pagamento para consulta do vendedor. O modo como o vendedor obtém esta informação, ou se é notificado da transacção, não está definido.

A arquitectura do SDD que inclui *e-Mandates* é composta pelos seguintes componentes: devedor, credor, banco do devedor, banco do credor, CSMs, serviço de encaminhamento, serviço de validação, directório e entidade certificadora.

- Devedor: entidade análoga ao cliente da SCT. Autoriza o débito directo da sua conta para a conta do credor. As cobranças que se seguem à autorização são descontadas da sua conta. Tal como na SCT não é especificado o dispositivo que o representa. No entanto, é sugerido que o devedor utiliza um *browser* para comunicar com as restantes entidades.
- Credor: entidade análoga ao vendedor da SCT. Recebe na sua conta os pagamentos debitados da conta do devedor.
- Banco do Devedor: componente análogo ao banco do cliente da SCT. É responsável pela conta do devedor.
- Banco do Credor: componente análogo ao banco do vendedor da SCT. É responsável pela conta do credor.
- CSMs ou Mecanismos de Compensação e Liquidação: não existem diferenças significativas em relação aos CSMs da SCT.
- Serviço de Encaminhamento: este componente tem como função obter, a pedido de um credor, um *e-Mandate* assinado pelo serviço de validação correspondente ao banco do devedor. O banco do credor pode incluir este serviço em vez de o delegar a terceiros.
- Serviço de Validação: esta entidade assina o pedido de débito com a devida autorização do devedor, criando um *e-Mandate*. O banco do devedor pode incluir este serviço em vez de o delegar a terceiros.
- Directório: o serviço oferecido por este componente é basicamente uma base de dados que, dado um BIC, devolve o URL (*Uniform Resource Locator*) do serviço de validação correspondente.
- Entidade Certificadora ou CA (*Certification Authority*): fornece os certificados necessários na comunicação do sistema e na verificação do e-Mandate. No *standard* são propostos certificados com extensões que indicam se uma entidade pode representar o papel de serviço de validação ou de serviço de encaminhamento. Apenas CAs pertencentes a um conjunto aprovado pelo EPC podem certificar serviços de validação ou encaminhamento. São utilizados certificados do lado do servidor em todas as comunicações. São ainda usados certificados do lado do cliente na comunicação entre serviços de validação e encaminhamento, assim como opcionalmente, na comunicação entre o credor e o serviço de encaminhamento.

Na Figura 2 está representada uma possível adaptação da arquitectura descrita a um sistema de pagamentos móveis. Aproveita-se também a figura para introduzir os conceitos associados a um sistema de pagamentos móveis.

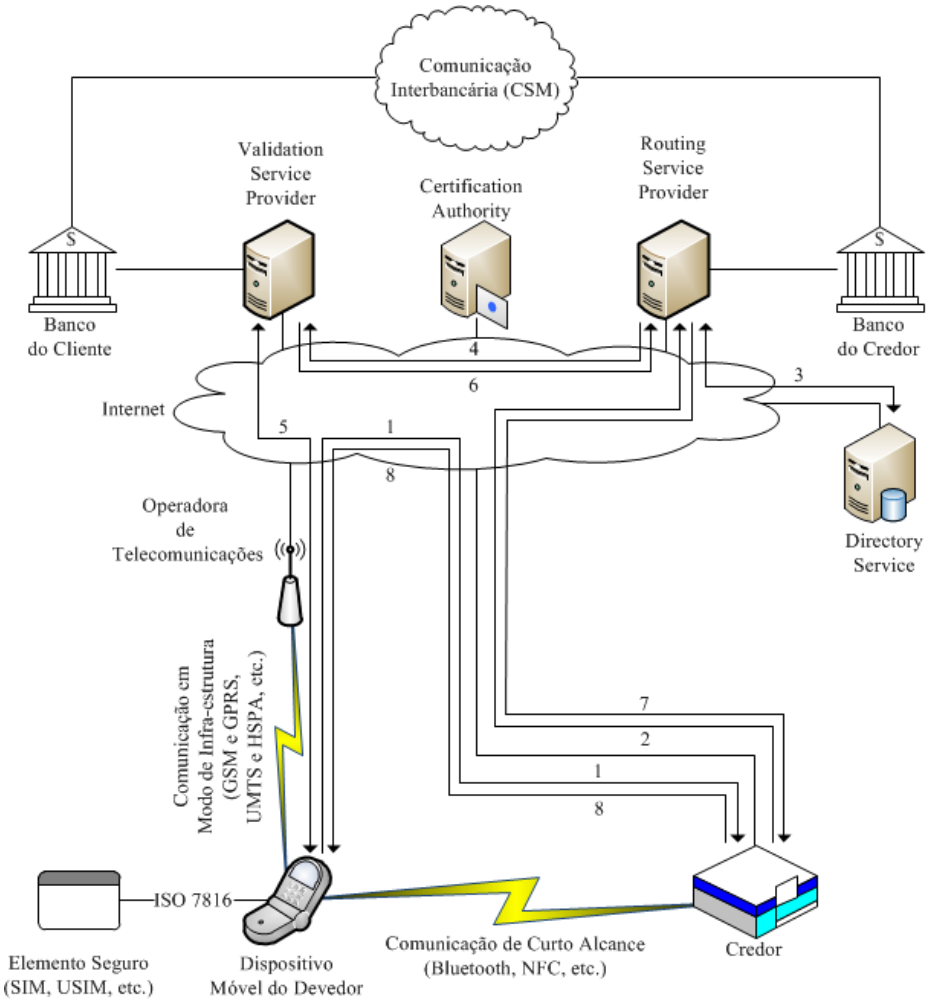


Figura 2. Adaptação da arquitectura do SDD de modo a representar os conceitos envolvidos num sistema de pagamentos móveis. Em relação à representação original, o componente Cliente passou a ser representado por um dispositivo móvel. Foram também incluídos o elemento seguro, a rede de acesso do dispositivo e uma ligação entre o devedor e o credor. O conceito de Comunicação em Modo de Infra-estrutura será abordado na secção 2.4 deste documento. O conceito de Comunicação de Curto Alcance será abordado na secção 2.5 deste documento. Num sistema de pagamentos móveis, esta ligação de curto alcance poderia ser utilizada para executar os passos 1 e 8.

Os procedimentos do SDD podem ser divididos em duas fases. A primeira corresponde à emissão do mandato (*e-Mandate*). A segunda fase corresponde a cada débito posterior.

A primeira fase pode ser resumida nos seguintes passos, representados na Figura 2:

- Passo 1. O devedor indica ao credor que lhe pretende passar a pagar através de débito directo. Para atingir este objectivo, o devedor entrega ao credor informação necessária para criar um e-mandate, incluído o BIC do banco do devedor. O *standard* sugere que o devedor preenche esta informação numa página *web* do credor.
- Passo 2. O credor cria um pedido de *e-Mandate*, no qual insere a sua informação, como o IBAN da sua conta. O pedido é enviado ao serviço de encaminhamento do banco do credor.
- Passo 3. O serviço de encaminhamento pede ao directório o URL do serviço de validação do banco correspondente ao BIC fornecido pelo devedor.

- Passo 4. O serviço de encaminhamento envia o pedido ao serviço de validação do banco do devedor. O serviço de validação verifica a autenticidade dos dados do pedido (como o IBAN do devedor).
- Passo 5. O serviço de validação autentica o devedor de acordo com o acordado com o banco do devedor. A forma como o faz não está definida no *standard.* Depois de devidamente autenticado, o devedor autoriza a adesão ao serviço de débito directo. Com o pedido autorizado, o serviço de validação assina o pedido, criando o objecto a que atribui o nome de *e-Mandate*.
- Passo 6. O serviço de validação retorna o *e-Mandate* ao serviço de encaminhamento, que verifica a assinatura do *e-Mandate*.
- Passo 7. O serviço de encaminhamento devolve o *e-Mandate* ao credor, que pode voltar a verificar a assinatura do *e-Mandate* ou confiar no serviço de encaminhamento.
- Passo 8. O credor guarda o *e-Mandate* e envia uma cópia ao devedor de modo a confirmar o sucesso da operação.

Ao obter o *e-Mandate*, o credor passa a estar autorizado a executar débitos directos da conta do devedor. Os passos para executar um débito directo não estão representados na Figura 2, para simplificar a figura. O processo é semelhante ao necessário para executar uma SCT, com uma diferença no primeiro passo. No SDD o primeiro passo tem origem no credor e não no devedor. O processo da transacção pode ser descrito pelos seguintes passos:

- Passo 0. Antes de executar o débito, o credor notifica o devedor sobre o pagamento.
- Passo 1. O credor envia um pedido de débito ao seu banco.
- Passo 2. O banco do credor comunica o pedido de débito ao banco do devedor através do CSM apropriado. O valor da conta do credor é actualizado.
- Passo 3. O banco do devedor desconta o valor do débito na conta do devedor.

2.2.3 Gestão de Aplicações em *Smartcards*

O EPC, em associação com a GSMA (GSM *Association*), pretende desenvolver um *standard* que considere pagamentos com origem em dispositivos móveis. No momento em que este documento foi escrito, esta intenção apenas se concretizou em dois documentos [13][14].

Estes documentos deixam antever suporte para sistemas de pagamento baseados em NFC. Nesta iniciativa, o NFC é visto como uma tecnologia que permite ao dispositivo móvel simular um cartão *contactless*. Esta visão tem como objectivo aproveitar ao máximo a infra-estrutura de pagamento por cartão de crédito ou debito. Estão também previstos os tipos de pagamento aplicáveis. Estão previstos os tipos de pagamentos remotos que serão abordados na secção 2.3.2. Estão também previstos os tipos de pagamentos de proximidade que serão abordados na secção 2.3.2, com excepção dos pagamentos P2P.

Apesar do processo da transacção em si não estar ainda especificado, no segundo documento [14] é explicada a abordagem ao ciclo de vida das aplicações de pagamentos móveis. Sobre esta gestão das aplicações são referidos os actores envolvidos, assim como as diversas acções que podem ser tomadas pelos mesmos, por exemplo, instalar, bloquear ou remover a aplicação. Sobre o processo da

transacção são sugeridos alguns dos actores envolvidos. É apresentado também um esboço dos passos que compõem um pagamento, adaptado dos pagamentos tradicionais por cartão de crédito ou débito.

Os actores que interagem em ambas as situações são: cliente, vendedor, banco do cliente, banco do vendedor, operadora móvel e TSM *(Trusted Service Manager).* O TSM foi criado com o objectivo de servir de intermediário entre várias operadoras e bancos. Para especificar as funções de cada um destes actores, com excepção do vendedor e do seu banco, a iniciativa referida utiliza a estrutura definida pela GlobalPlatform. A GP (GlobalPlatform) define um actor adicional, o CKLA (*Confidential Key Loading Authority*). Este componente permite a configuração de chaves no *smartcard* do dispositivo do cliente de forma confidencial. A arquitectura da GP considera ainda o fornecedor de *smartcards*.

A GP é uma associação internacional, cujo objectivo é manter a interoperabilidade entre as entidades envolvidas na gestão de aplicações em *smartcards*. A GP especifica a interacção entre estes actores na instalação, modificação e remoção de aplicações [15]. Deste modo é possível ter no mesmo *smartcard* várias aplicações pertencentes a entidades distintas.

Para representar os processos envolvidos na gestão das aplicações, a GP define os papéis a serem desempenhados. A distribuição destes papéis pelos actores envolvidos não está especificada. Um papel pode ser desempenhado em conjunto por vários actores e um actor pode desempenhar vários papéis em simultâneo. Os papéis definidos pela GP são os seguintes: *Application Developer*, *Application Owner*, *Application Provider*, *SSD (Supplementary Security Domain) Manager*, *Controlling Authority*, *Card Issuer*, *Cardholder*, *Card Enabler*, *Loader*, *Card Manufacturer*, *IC (Integrated Circuit) Manufacturer*, *Platform Specification Owner* e *Platform Developer*. De seguida, são descritos estes papéis. Alguns foram agrupados de modo a simplificar a sua descrição.

- *Application Developer/Owner*: desenvolve e é responsável pelo código da aplicação que deve ser instalada no *smartcard*.
- *Application Provider:* Prepara a aplicação para ser carregada no *smartcard* com os dados necessários. Utiliza a aplicação para oferecer um serviço ao *Cardholder*.
- SSD *Manager*: Possui chaves que lhe permitem comunicar indirectamente com o *smartcard* de forma confidencial. Este papel é desempenhado pelo actor que está incumbido de controlar a gestão da aplicação.
- *Controlling Authority*: Este papel representa de forma directa a função do CKLA, ou seja, permite instalar chaves no *smartcard* de forma confidencial, para que possam ser usadas em outras comunicações.
- *Card Issuer*: Responsável pelos cartões que distribui pelos clientes. Controla toda a gestão do cartão até ao momento da sua entrega ao cliente. Depois da entrega, pode controlar total ou parcialmente o acesso ao cartão. Gere que aplicações têm permissão para serem instaladas nos cartões.
- *Cardholder*: entidade que usufrui do cartão com a autorização do *Card Issuer*.
- *Card Enabler*: Inicializa o conteúdo do cartão.

- *Loader*: Comunica com o cartão através da sua infra-estrutura, executando comandos a pedido de outras entidades.
- *Card Manufacturer*/IC *Manufacturer/Platform Specification Owner/Platform Developer:* Providencia os cartões com o respectivo sistema operativo às restantes entidades. Na Figura 3 este conjunto de papéis é denominado *Card Provider*, por forma a ocupar menos espaço.

A GP exemplifica a atribuição dos papéis aos actores através de vários cenários. Estes cenários podem ser resumidos em três tipos: modo simples, modo de delegação e modo de autorização.

Modo simples: Neste modo a operadora mantém controlo parcial ou total sobre o acesso ao cartão. Ao TSM cabe apenas a verificação da integridade da aplicação. O banco do cliente, ou *Service Provider* como é denominado pela GP, delega toda a gestão das aplicações ao TSM.

Modo de delegação: Neste modo a operadora abdica do controlo sobre o acesso ao cartão. Assim, cabe ao TSM instalar, modificar e remover aplicações do *smartcard*, através da sua plataforma OTA (*Over The Air*). A tecnologia OTA permite aceder remotamente ao *smartcard* do dispositivo móvel. Apesar deste controlo, o TSM tem de pedir permissão à operadora sempre que desejar aceder ao *smartcard*. O *Service Provider* pode delegar toda a gestão das aplicações ao TSM, ou pode não delegar a modificação das aplicações. No segundo caso, o *Service Provider* prepara um comando de modificação cifrado e entrega-o ao TSM, de modo a que este envie o comando ao *smartcard* sem ler o seu conteúdo. O comando é decifrado no *smartcard*.

Modo de autorização: Neste modo a operadora também abdica do controlo sobre o acesso ao cartão. A diferença para o modo anterior reside na liberdade dada ao TSM para aceder ao *smartcard*. No modo de autorização o TSM não necessita de pedir permissão à operadora para cada operação.

A Figura 3 ilustra um exemplo de uma possível arquitectura de gestão de aplicações de pagamentos móveis, com base nas indicações do EPC, da GSMA e da GP abordadas anteriormente. O cenário do exemplo enquadra-se no modo simples.

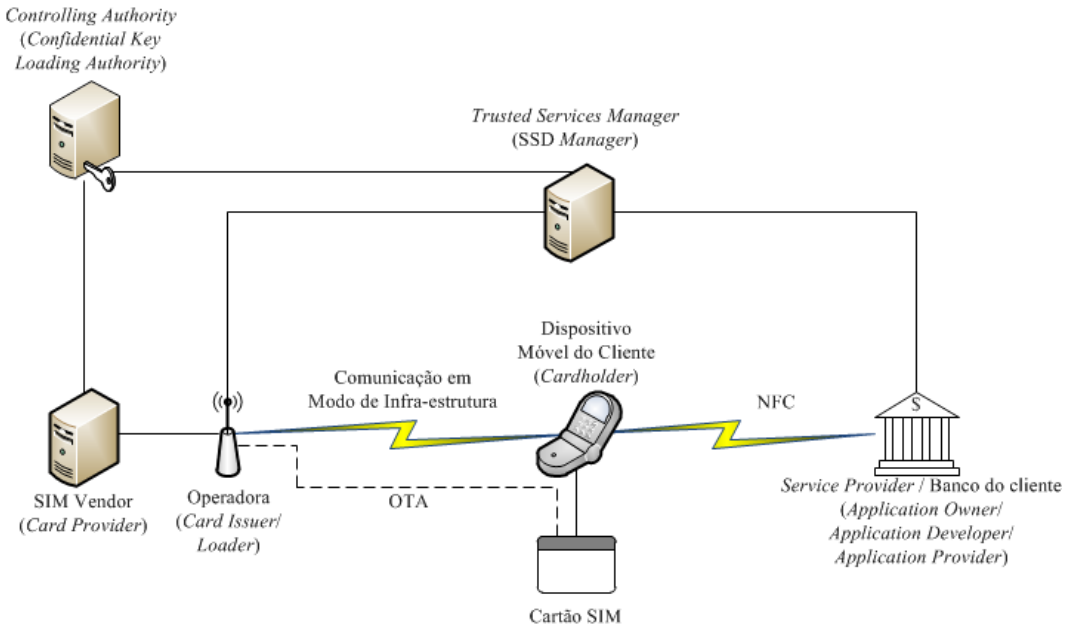


Figura 3. Exemplo de uma arquitectura de gestão de uma aplicação de pagamentos móveis. Foi incluída uma ligação NFC entre o dispositivo do cliente e o banco porque, num dos documentos referidos [14], é sugerido que a gestão da aplicação poderia ser executada desta forma.

2.3 Tipos de Sistemas de Pagamento Móvel

Apesar da elevada quantidade e diversidade de sistemas desenvolvidos na área dos pagamentos móveis, estes podem ser caracterizados em categorias que representam os seus principais atributos. Esta classificação é exposta nesta subsecção do relatório.

2.3.1 Valor Monetário das Transacções

Um pagamento móvel pode ser classificado pela quantia envolvida. Existem várias delimitações de categorias, variando no número de divisões e no valor dos limites. O sistema *Paycircle*, por exemplo, define dois tipos de pagamento, separados pela barreira dos €10. Neste documento adopta-se a definição do *European Committee for Banking Standards*, que considera as categorias seguintes [3].

- Micro pagamento: pagamentos até €2. Este tipo de pagamentos tem como principal concorrente o dinheiro em papel [5]. Numa destas transacções a segurança pode ser um pouco mais relaxada. No entanto, preocupações como rapidez, anonimato e custo ganham maior relevância.
- Mini pagamento: pagamentos desde €2 até €25. Assim como os seus valores, também as características deste tipo de pagamentos são um meio-termo entre os micro e macro pagamentos.
- Macro pagamento: pagamentos desde €25. Este tipo de transacção electrónica tem como principal concorrente o cartão de crédito [5]. As principais preocupações contrastam, obviamente, com as dos micro pagamentos. À medida que a quantia do pagamento aumenta, o custo da transacção torna-se desprezável em relação ao valor do pagamento. Por outro lado, requisitos como a segurança do pagamento tornam-se mais importantes do que, por exemplo, a rapidez da transacção.

2.3.2 Tipo de Interacção

As diferentes utilizações dos pagamentos móveis [16] podem ser categorizadas em dois tipos: pagamentos à distância e pagamentos de proximidade. Estas categorias são descritas de seguida.

Pagamentos à distância: Neste tipo de pagamentos o utilizador recebe a informação da transacção remotamente, geralmente através de uma página Web que inicia um pedido de pagamento. Esta situação, em que o cliente interage com um vendedor à distância, denomina-se ponto de venda virtual ou POS (*Point of Sale*) virtual. Este é o caso de páginas de compra de produtos, sendo o exemplo mais vulgar a personalização de telemóveis (*download* de toques, imagens de fundo, jogos, etc.). Em outras situações o utilizador já conhece a informação necessária *a priori*. Este é o caso das tradicionais transferências bancárias, em que o utilizador sabe o NIB do destinatário, ou os códigos de entidade e referência correspondentes ao pagamento. Os pagamentos C2C (*customer to costumer*) remotos constituem outro exemplo que também se pode incluir nesta categoria.[1] Os pagamentos à distância pressupõem a utilização de tecnologias de comunicação em modo de infra-estrutura, algumas das quais serão referidas posteriormente neste documento.

Pagamentos de proximidade: Neste tipo de pagamento o dispositivo do utilizador interage directamente com o destinatário. Poderá ou não existir comunicação com um servidor durante a transacção. Esta categoria engloba situações como pontos de venda (POS), P2M (pagamentos em maquinas automáticas) ou P2P (*peer to peer*). O ponto de venda representa a transacção tradicional entre o cliente e o vendedor num estabelecimento comercial. A utilização de pontos de venda assume a modificação do terminal do vendedor para ser compatível com o sistema de pagamentos móveis. O P2M corresponde a um caso específico de ponto de venda, em que o processo tem de estar automatizado e o valor do pagamento é em geral reduzido. O P2P representa uma situação em que não existe um vendedor, que utiliza um equipamento diferente. Numa transacção deste tipo ambos os intervenientes têm o mesmo tipo de equipamento (dispositivo móvel). Para além das tecnologias de comunicação em modo de infra-estrutura, esta categoria usa também tecnologias de comunicação de curto alcance, cujos exemplos serão ilustrados na secção correspondente deste relatório.

2.3.3 Momento do Pagamento

O momento em que são cobradas as transacções é uma consequência directa da arquitectura escolhida. Existem as seguintes alternativas [16]:

- Em tempo real: Nos sistemas em que a transacção implica uma comunicação imediata com o servidor, é possível efectuar operações verificar o saldo e transferir fundos durante a transacção. Se não existir saldo suficiente a operação pode ser canelada.
- Pré-pago: Assim como nos tarifários pré-pagos da telefonia móvel, o cliente “carrega” o dispositivo antes de poder executar transacções. Quando o cliente fica sem saldo, tem de recarregar o dispositivo para voltar a poder efectuar pagamentos.

[1] Este tipo de pagamentos é por vezes considerado P2P (*peer to peer*). No entanto neste documento um pagamento é considerado P2P se existir interacção directa entre os dois clientes, o que não é o caso.

- Pós-pago: O cliente paga pelo serviço depois de o utilizar. Em alguns sistemas o cliente paga periodicamente pela soma das transacções que executou. Outros sistemas funcionam de forma semelhante aos cheques, isto é, o valor da transacção é transferido quando o destinatário "deposita" a transacção.

2.3.4 Tipo de Transacção

No cerne de um serviço de pagamentos electrónicos encontra-se o conceito de transacção. A definição deste conceito agrupa os sistemas em duas categorias, influenciadas pelas alternativas disponíveis em pagamentos tradicionais.

***Account-based* [17]:** Esta categoria não deve ser confundida com sistemas baseados em contas bancárias. Uma solução *account-based* pode estar associada directamente a uma conta bancária, contudo nada a impede de estar associada a um cartão de crédito, ou a um contracto com uma operadora ou outra entidade. Este tipo de sistema é análogo aos cheques, no sentido em que o utilizador assina um documento, que contém a informação da transacção como valor do pagamento e a identidade do destinatário. O documento é posteriormente entregue à instituição responsável por concretizar a transferência. Num pagamento móvel a transacção é representada por uma mensagem que contém a informação necessária para identificar o pagamento. A informação pode ser formada pelos campos da transacção, à semelhança de um cheque, ou por um resumo dos mesmos, geralmente criado através de funções de *hashing* [6][18]. Esta última opção tem como vantagens a não representação dos campos em claro, assim como a possível redução do tamanho da mensagem. Assim como nos cheques, a prova da transacção é uma assinatura, neste caso uma assinatura digital. Em muitos sistemas esta assinatura baseia-se no método tradicional, utilizando criptografia assimétrica geralmente acompanhada por certificados digitais [6]. No entanto, alguns dos sistemas não aderem completamente à ideia tradicional de assinatura digital, como é o caso dos baseados em criptografia simétrica [19]. A complexidade deste tipo de pagamentos concentra-se no instante da transacção, momento em que se processa a maior parte das operações mais complexas, como algoritmos criptográficos.

Baseado em *tokens* [5]: Esta solução é comparável à utilização de moedas e notas. Os cartões de débito pré-pagos são outro exemplo deste tipo de sistema [20]. Tal como nos exemplos referidos, o objecto que representa a transacção não é criado durante a mesma. No momento em que estes pagamentos são efectuados o cliente já possui um objecto válido, criado *a priori* com o aval de uma entidade confiável como uma instituição bancária. Os protocolos baseados em *tokens* são constituídos por 3 fases. A primeira fase representa um levantamento de dinheiro ou carregamento de um cartão. Nesta fase o cliente adquire os *tokens* que lhe permitem executar pagamentos. A fase correspondente à transacção é, em teoria, mais simples do que na opção *account-based*. Esta fase pode ser resumida a uma troca de *tokens* entre os participantes, seguida de uma verificação da validade dos mesmos por quem os recebe. Na prática, esta troca costuma incluir uma prova do pagamento e da identidade dos participantes, o que aumenta a complexidade da operação. A última fase representa um depósito, constituído pela entrega dos *tokens* recebidos em transacções à entidade central do sistema.

Os protocolos de sistemas baseados em *tokens* são compostos por mais fases que os dos sistemas *account-based*. A fase adicional que representa um levantamento obriga o utilizador a "carregar" o dispositivo com *tokens*, constituindo assim a principal desvantagem desta opção. Por outro lado, a facilidade de garantir anonimato, assim como um custo por transacção inferior [17], podem ser vistas como vantagens em relação aos sistemas *account-based*. No entanto, a propriedade de anonimato também pode ser implementada neste tipo de sistemas [21] e o custo por transacção pode ser reduzido, através da agregação de micro e mini pagamentos [17][22].

2.3.5 Necessidade de Intermediários

Na grande maioria dos sistemas que implementam pagamentos de proximidade pode encontrar-se um denominador comum. Ao efectuar uma transacção, independentemente do tipo de interacção escolhido (POS, P2P, etc.), existe uma interacção obrigatória com uma entidade central. Estas transacções podem ser categorizadas como *online*. Esta comunicação de longo alcance introduz um conjunto de problemas num serviço de pagamentos móveis [23]. Para além de aumentar o consumo energético do dispositivo, um recurso valioso num ambiente móvel, este processo torna a transacção mais lenta e mais cara. Tornam até impossível a utilização do sistema, em zonas sem cobertura da rede telefónica móvel.

Se o tipo de interacção incluir um cliente e um vendedor, estes problemas podem ser minimizados. A comunicação com a entidade central pode ser feita exclusivamente através do equipamento terminal do vendedor. Por não ser um dispositivo móvel, o consumo energético não é um factor crítico. Por outro lado, a velocidade e o preço da ligação são mais favoráveis numa rede fixa.

Todavia, num pagamento P2P ambos os intervenientes utilizam dispositivos móveis. Neste tipo de pagamento os problemas referidos são evitados em sistemas cujas transacções não necessitem de comunicação imediata com uma terceira entidade. Estes serviços podem ser classificados como *offline*. Porém, estes sistemas não estão livres de desvantagens, visto terem de lidar com problemas que derivam da falta de controlo da entidade central durante a transacção.

Em sistemas *Account-based,* nos pagamentos *offline* existe a possibilidade de o cliente efectuar pagamentos para os quais não tem saldo. Em soluções de transacções com intermediários este problema não existe, porque o saldo do cliente é verificado durante a transacção. Este problema é análogo a passar um cheque sem cobertura.

No caso dos sistemas baseados em *tokens*, os pagamentos *offline* dão origem a um problema distinto. Como na fase de levantamento é possível verificar o saldo do cliente, a utilização devida dos *tokens* levantados não ultrapassará o saldo disponível. O problema destes sistemas prende-se com a possibilidade de reutilização indevida de *tokens*. Ao contrário dos objectos físicos análogos ao *token* (moedas ou notas), ao entregar um *token*, o cliente não deixa automaticamente de o possuir. Logo, o cliente pode usar o mesmo *token* em vários pagamentos, efectivamente multiplicando dinheiro. Sem nenhum método auxiliar, o receptor dos *tokens* apenas verifica a validade dos mesmos, que continuam válidos porque foram gerados com o consentimento da entidade responsável. O problema

apenas é detectado no momento dos depósitos, visto que o banco recebe várias vezes o mesmo *token*.

Apesar dos *tokens* duplicados serem facilmente detectados, a verdadeira dificuldade centra-se em encontrar o autor da duplicação. Esta dificuldade é ainda agravada no caso dos *tokens* forem transferíveis, isto é, se entre o levantamento e depósito na entidade responsável, o *token* circula por mais de duas entidades. Neste caso, a lista de culpados engloba todas as entidades pelas quais o *token* passou. Para responder a este problema, os autores de sistemas deste género tomam dois tipos de medidas.

O primeiro tipo de medidas consiste em utilizar um meio de armazenamento e processamento seguro, como é o caso do cartão SIM dos telemóveis. Nos sistemas que usam esta medida, os autores depositam a sua confiança no meio seguro, assumindo que este componente do dispositivo do cliente segue correctamente os passos do protocolo. Assumem também que a segurança deste componente é inviolável.

O segundo tipo consiste em manter mais informação sobre o percurso de cada *token*, de modo a conseguir detectar a origem da duplicação. Esta informação é mantida de forma distribuída, em cada dispositivo pelo qual o *token* passa e é análoga ao conceito de recibo. Ao transferir um *token*, os intervenientes guardam um registo sobre quem enviou e quem recebeu o *token*. As garantias de anonimato e integridade destes registos variam de sistema para sistema. Alguns sistemas de *tokens* não transferíveis obrigam ainda o cliente a provar que levantou o *token*, de modo a demonstrar que não está apenas a retransmiti-lo.

Para minimizar o problema dos *tokens* transferíveis, alguns sistemas tomam medidas adicionais, como limitar o número de saltos que um *token* pode dar até ter de ser depositado. O sistema fairCASH [24] é um exemplo de um sistema que adopta esta medida.

2.3.6 Tipo de Criptografia

A escolha do tipo de criptografia, usado para cumprir os requisitos de segurança de um sistema de pagamentos móveis, altera as características do sistema. Devido à importância da garantia de não repudiação de um pagamento, muitos dos trabalhos na área dos pagamentos móveis utilizam criptografia assimétrica. Este tipo de criptografia facilita a implementação de não repudiação através da utilização de assinaturas digitais, cujo valor legal é reconhecido em diversos países incluindo na União Europeia [25][6]. A ideia de fornecer não repudiação com esta técnica assenta na aplicação de um segredo, conhecido somente pelo emissor, a uma mensagem. Se o receptor conseguir determinar inequivocamente que a mensagem foi gerada através do dito segredo, mesmo sem o conhecer, consegue determinar também o autor da mensagem.

A dificuldade em atingir este objectivo, ao adoptar criptografia simétrica, prende-se com o facto de, em contraste com a criptografia assimétrica, o segredo usado nos algoritmos criptográficos ser partilhado pelos intervenientes. Logo, não é possível provar que um utilizador em específico foi o autor de uma mensagem. No entanto, soluções com base em criptografia simétrica também podem oferecer esta garantia, desde que não exista conluio entre um vendedor e a entidade central do

sistema. Para o conseguir, o sistema usa vários segredos partilhados com entidades diferentes. O emissor do pagamento aplica à mensagem um segredo partilhado apenas com o receptor e, de seguida, aplica um segredo partilhado somente com a entidade central. Cada um dos outros intervenientes conhece um dos segredos, porém apenas o emissor do pagamento conhece os dois. Logo, agregando a informação do receptor do pagamento e da entidade central é possível garantir que o emissor autorizou uma transacção.

A capacidade computacional necessária para processar criptografia simétrica é inferior à necessária em criptografia assimétrica. Esta revela-se como a principal vantagem da criptografia simétrica [26], acentuada pelo contexto de um ambiente móvel, em que os recursos são especialmente limitados. A criptografia simétrica introduz contudo uma desvantagem nos sistemas de pagamentos móveis. Estes protocolos incluem geralmente uma fase anterior à transacção, que deve ser executada para cada dispositivos ao qual se deseja efectuar pagamentos. Esta fase de registo é composta pela negociação de um conjunto de chaves que serão usadas em transacções posteriores.

Existem ainda sistemas mistos que combinam a utilização de criptografia simétrica e assimétrica, de forma a equilibrar as vantagens e desvantagens de ambas. Uma das formas de aplicar as duas técnicas consiste em usar criptografia assimétrica para assinar mensagens, de forma a garantir não repudiação, utilizando contudo criptografia simétrica para as cifrar, com o objectivo de melhorar a performance do sistema.

2.4 Tecnologias de Comunicação em Modo de Infra-estrutura

Todos os sistemas de pagamentos móveis necessitam de comunicar com um servidor de qualquer tipo. Em alguns tipos de sistema, o dispositivo móvel interage com o servidor para levantar ou depositar créditos. Em outros sistemas a interacção ocorre no momento da transacção. No entanto, em qualquer sistema, esta interacção é inevitável. Para manter a ubiquidade do sistema, esta comunicação tem de ser feita em modo de infra-estrutura, ou seja, através de uma rede que permita interagir com o servidor a grandes distâncias. Deste modo o cliente pode interagir com o sistema em qualquer local em que a rede do operador móvel tenha cobertura. De seguida enumera-se algumas das tecnologias usadas para este fim [27].

2.4.1 GSM

O GSM (*Global System for Mobile Communications*) é o sistema de comunicação para redes telefónicas móveis mais usado do mundo (com mais de 3 mil milhões de dispositivos [28]). Apesar de este sistema fornecer alguma segurança aos serviços implementados sobre o mesmo, apresenta também algumas vulnerabilidades que não podem ser ignoradas [22]. O GSM garante apenas confidencialidade da comunicação entre o dispositivo móvel e a *base station*, visto que esta é a única parte da rede em que os dados se encontram cifrados. A segunda vulnerabilidade prende-se com a inexistência de autenticação da *base station* perante o dispositivo móvel, o que permite a atacantes fazerem-se passar por *base stations* e executarem ataques *man-in-the-middle* [29].

Estas características têm de ser tidas em conta quando se desenha um serviço sobre uma rede GSM, visto que se o serviço necessitar de oferecer garantias como as referidas (confidencialidade e autenticação mútua) terá de as implementar.

2.4.2 SMS e USSD

O SMS (*Short Message Service*) utiliza o canal de sinalização do GSM para transportar mensagens de texto até 160 caracteres. Este serviço surgiu aquando do lançamento da segunda geração de telemóveis, o que o torna num dos mais antigos serviços das redes de telefonia móvel. Esta universalidade, em conjunto com a sua popularidade [30][31] e simplicidade levou-o a ser adoptado por muitas das soluções para pagamentos móveis.

Contudo, as soluções baseadas em SMS têm limitações, principalmente na área da segurança, o que restringe geralmente esta tecnologia a soluções de pagamentos de pequenas quantias [6] [7][22]. Um dos problemas do SMS é não garantir confidencialidade ou integridade das mensagens. Para além dos problemas herdados do GSM, que tornam possível interceptar, ler e alterar a mensagem enquanto é transmitida pela rede, esta é guardada em claro no dispositivo móvel. Outra desvantagem centra-se na inexistência de autenticação do utilizador perante o telemóvel no acto de envio de uma mensagem, o que permite a qualquer pessoa que se apodere do telemóvel efectuar pagamentos em nome do dono do dispositivo. Esta vulnerabilidade dificulta a garantia de não repudiação, o que faz com que o cliente possa negar que executou qualquer um dos pagamentos, visto que nenhuma entidade do sistema pode provar o contrário.

O USSD (*Unstructured Supplementary Service Data*) [27] é um serviço suportado em GSM semelhante ao SMS, com a diferença de ser orientado à sessão, o que pode originar melhores tempos de resposta em comunicações bidireccionais [29]. Este protocolo não apresenta melhorias significativas em relação às desvantagens referidas do SMS.

2.4.3 GPRS

O GPRS (*General Packet Radio Services*) [27] é a tecnologia de troca de pacotes de dados sobre canais de voz do sistema GSM, em contraste com a comutação de circuitos usada nos restantes serviços do GSM, ou do SMS e USSD que utilizam o canal de sinalização. Esta tecnologia de transmissão de dados serve como base de vários serviços como MMS (*Multimedia Messaging Service*) e WAP (*Wireless Application Protocol*). A velocidade de transmissão do GPRS pode, teoricamente, atingir um máximo de 171 Kbit/s [32]. Por estas razões esta tecnologia é utilizada como uma alternativa ao USSD e ao SMS. As principais vantagens em relação ao USSD e ao SMS centram-se nos débitos oferecidos e no facto de esta ser uma tecnologia direccionada para a transmissão de dados genéricos, em vez de mensagens curtas de texto, como é o caso das alternativas referidas.

2.4.4 UMTS e HSPA

O UMTS (*Universal Mobile Telecommunications System*) [33] é um sistema que visa substituir o GSM como sistema de comunicação para redes de telefónicas móveis. O UMTS pertence à terceira geração de telefonia móvel, que inclui serviços como videochamadas.

O HSPA (*High Speed Packet Access*) [34] é uma tecnologia de troca de pacotes desenvolvida sobre UMTS. Esta tecnologia é utilizada em serviços de banda larga móvel, visto que atinge velocidades até 14 Mbit/s no sentido descendente e 5.8 Mbit/s no sentido ascendente.

2.4.5 LTE

O LTE (*Long Term Evolution*) [35] afigura-se como a tecnologia de suporte das redes telefónicas móveis de quarta geração. O LTE suporta larguras de banda entre 1.4 e 20 MHz, atingindo velocidades 100 Mbit/s no sentido descendente e 50 Mbit/s no sentido ascendente. A principal vantagem desta tecnologia, em relação a alternativas como o WiMax [36], é a compatibilidade com as tecnologias anteriores (GSM e UMTS).

2.5 Tecnologias de Comunicação de Curto Alcance

Em sistemas de pagamentos móveis a segurança é uma das prioridades que mais se destacam. Esta preocupação leva a que os critérios, que determinam a escolha de tecnologias de comunicação de curto alcance, sejam algo diferentes em relação a outro tipo de sistemas. Um exemplo desta diferença de critérios pode verificar-se na análise do alcance das tecnologias. Em outras áreas, alcances elevados ou a possibilidade de comunicação em linha de vista são encaradas como vantagens. No entanto, num sistema de pagamentos móveis essas características são vistas como uma redução da privacidade da comunicação e, consequentemente, como desvantagens da tecnologia.

2.5.1 IrDA

IrDA (*Infrared Data Association*) permite comunicação entre 2 dispositivos em linha de vista a uma distância até 1 metro, atingindo débitos até 4.0 Mbit/s. Para este efeito é utilizada a zona do espectro correspondente a infravermelhos (comprimentos de onda entre 850 e 900nm)[37].

As principais vantagens desta tecnologia são a facilidade de configuração e uso, o elevado número de dispositivos que já a suportam e o consumo reduzido de energia [38].

2.5.2 Bluetooth

O *Bluetooth* [39] é uma das tecnologias de comunicação de curto alcance mais usadas nos pagamentos móveis. Opera na banda dos 2.4 GHz e atinge velocidades até 2 Mbit/s (na versão 2.0 do protocolo). O alcance da tecnologia varia aproximadamente entre 1 e 100 metros em função da potência de transmissão escolhida. O *Bluetooth* partilha algumas vantagens com o IrDA,

nomeadamente a universalidade e o consumo de energia. A estas mais-valias junta-se ainda uma preocupação com a segurança da comunicação, mais especificamente com confidencialidade e autenticação dos dispositivos. Esta preocupação manifesta-se através da utilização de cifra com chaves de 128 bits, assim como um processo de emparelhamento, executado na fase inicial da primeira comunicação entre dois dispositivos. Este processo consiste na utilização de um segredo conhecido à partida pelos dois dispositivos, geralmente introduzido manualmente pelos utilizadores, com o objectivo de gerar uma chave partilhada por ambos. Contudo, a segurança do protocolo é geralmente considerada insuficiente pelos sistemas de pagamento móvel, o que os leva a implementar medidas de segurança extra sobre o *Bluetooth* [6]. São inclusivamente conhecidas vulnerabilidades no protocolo que podem levar a ataques como *man-in-the-middle* [40][29]. O processo de emparelhamento é também por vezes visto como uma parte negativa do protocolo por atrasar estabelecimento da ligação [23].

2.5.3 RFID

A tecnologia RFID (*Radio-frequency identification*) é usada com sucesso em áreas como identificação de clientes em portagens, ou na segurança de lojas de vestuário. Um sistema RFID usa campos electromagnéticos para comunicar em frequências tão díspares como 125 KHz e 5,8 GHz [38]. Dependendo da frequência e tipo de RFID, o alcance pode variar entre alguns centímetros e 100 metros [41]. O consumo de energia também é influenciado pelo tipo de RFID. Um dispositivo RFID pode ser passivo, activo ou misto. Um dispositivo RFID passivo não dispõe de alimentação própria, sendo alimentado pelo sinal do emissor. Um dispositivo RFID activo inclui alimentação própria, compensando o custo energético com um alcance maior, visto que a potência do sinal não tem de ser elevada o suficiente para alimentar o receptor. Um dispositivo RFID misto tem alimentação própria, no entanto usa-a apenas para alimentar os circuitos do dispositivo, utilizando o sinal do emissor para alimentar a comunicação. Estes dispositivos têm um alcance intermédio entre os passivos e activos. Um sistema RFID consume ainda menos energia que as tecnologias anteriores, especialmente se for do tipo passivo.

A tecnologia tem, no entanto, dificuldades em situações de elevadas concentrações de dispositivos [42]. Esta desvantagem prende-se com duas deficiências inerentes à tecnologia. A primeira deficiência deve-se à dificuldade que um leitor RFID tem em interagir com vários receptores próximos entre si. A segunda deficiência reside na possibilidade de dois leitores RFID causarem interferências entre si.

2.5.4 NFC

NFC (*Near Field Communication*) [42][43] é uma tecnologia relativamente recente, baseada e compatível com RFID. O NFC opera na frequência dos 13,56 MHz e permite velocidades até 424 Kbit/s. Os dispositivos NFC podem funcionar em três modos.

- Em modo P2P, comunicando com outro dispositivo NFC.
- Como leitores de etiquetas NFC.

- Em modo de simulação de uma etiqueta NFC, para que um leitor o encare como um simples cartão *contactless*.

Um dispositivo compatível com NFC tem de incluir os seguintes componentes: uma antena NFC, um chip NFC e um elemento seguro.

- Antena: responsável por assegurar a comunicação com outros dispositivos.
- Chip NFC: componente central da tecnologia. Assegura a ligação da antena ao elemento seguro e ao resto do dispositivo.
- Elemento seguro: guarda dados e executa aplicações de forma segura. Pode existir mais do que um elemento seguro por dispositivo.

O elemento seguro é geralmente representado por um *smartcard* com suporte para a plataforma Java Card. O Java Card é um subconjunto da plataforma Java destinado a dispositivos com recursos escassos, como é o caso do *smartcard*. Tal como a plataforma Java, a Java Card tem como principal vantagem a portabilidade de aplicações entre *smartcards* diferentes. O elemento seguro pode ser implementado de várias formas. As principais opções são a utilização de um cartão SIM, ou de um elemento seguro embebido. Estas opções não são mutuamente exclusivas. Um telemóvel pode ter vários elementos seguros dos vários tipos. As possibilidades referidas são descritas de seguida.

- Cartão SIM: o elemento seguro é representado pelo cartão SIM do telemóvel. O cartão SIM é ligado ao chip NFC através de um fio único, segundo a especificação SWP (*Single Wire Protocol*) definida pelo ETSI (*European Telecommunications Standards Institute*). Esta opção é apoiada pelas operadoras de telecomunicações, visto serem as donas do cartão.
- Elemento seguro embebido: nesta opção, o elemento seguro é embebido no telemóvel, o que o torna independente da operadora. No entanto, a portabilidade da aplicação é reduzida, visto que não é possível transportar o elemento seguro de um telemóvel para outro. As instituições bancárias preferem esta opção.

O NFC inclui protocolos de comunicação que abrangem os níveis 1 e 2 da pilha de protocolos OSI (*Open Systems Interconnection*). Para a camada de protocolos inferior é definido um protocolo de comunicação denominado NFCIP-1 especificado no ISO 18092. Como alternativa a este protocolo, podem ser usados os protocolos ISO 14443 ou o FeliCa. O ISO 14443 é um *standard* genérico para comunicação com cartões *contactless*. As suas mensagens são baseadas no ISO 7816-4 referido anteriormente. O FeliCa é um protocolo baseado no ISO 18092 e adoptado principalmente no mercado asiático. Os três protocolos referidos resolvem aspectos da camada do nível 1 do modelo OSI como a modelação utilizada. No entanto, também englobam a parte inferior da camada do nível 2, por endereçarem o problema controlo de acesso ao meio ou MAC (*Media Access Control*).

No modo de comunicação P2P, também é especificada a parte superior da camada de nível 2 do modelo OSI. O LLCP (*Logical Link Control Protocol*) desempenha funções como permitir ligações simultâneas de vários protocolos de níveis superiores. Este modo de comunicação suporta protocolos de camadas superiores como TCP/IP (*Transmission Control Protocol/Internet Protocol*) ou OBEX (*OBject EXchange*).

Para além de herdar os baixos consumos do RFID, a principal característica do NFC é o alcance reduzido (3 a 30cm). Este factor torna intrusões extremamente difíceis, o que por sua vez torna desnecessários protocolos como o emparelhamento do *Bluetooth*. Assim, o estabelecimento de uma ligação é mais simples e rápido [38].

A desvantagem da utilização de NFC, em detrimento das tecnologias referidas anteriormente, centra-se no facto de, no momento em que este relatório foi escrito, esta tecnologia estar ainda pouco difundida. Contudo, está previsto um aumento dos dispositivos e sistemas operativos compatíveis com esta tecnologia [44][45].

2.6 Soluções Existentes

A presente subsecção tem como objectivo exemplificar várias formas de conceber um sistema de pagamentos móveis, através da descrição de alguns projectos desta área. A subsecção termina com uma tabela que classifica os projectos descritos em função das categorias referidas anteriormente (Tabela 1).

2.6.1 SEMOPS

O SEMOPS (*SEcure MObile Payment Service*) [46] é um projecto europeu iniciado em 2002, baseado na cooperação entre bancos e operadoras. O sistema é um dos mais complexos e completos, visto que disponibiliza um variado leque de aplicações, incluindo tipos de interacções referidos anteriormente como POS, POS virtual, P2P e P2M. É também compatível com várias tecnologias de comunicação. O cliente interage directamente com a entidade em que confia, denominada *payment processor*, que pode ser uma instituição financeira ou uma operadora.

São possíveis transacções das três categorias de valor monetário mencionadas (micro, mini e macro pagamentos), assim como pagamentos internacionais. O protocolo do SEMOPS inclui, contudo, comunicação obrigatória entre os utilizadores (emissor e receptor do pagamento) e uma entidade central, o que não permite pagamentos *offline*. No caso do *payment processor* ser um banco, os micro pagamentos podem também tornar-se caros, porque cada pagamento representa uma transacção bancária e é taxado como tal. O SEMOPS não especifica todos os pontos da arquitectura, com o objectivo de maximizar a liberdade das entidades que implementam o protocolo, mantendo simultaneamente o requisito da interoperabilidade. Este facto leva a que não estejam descritos determinados aspectos do protocolo, tais como tecnologias de comunicação a serem usadas entre os clientes, vendedores e *payment processors.*

A arquitectura do sistema engloba os seguintes componentes:

- *Customer*: Representa o cliente que emite o pagamento.
- *Merchant*: Vendedor que recebe o pagamento.
- CPP *(Customer's Payment Processor)*: Entidade responsável pela informação financeira do cliente.
- MPP *(Merchant's Payment Processor)*: Entidade responsável pela informação financeira do vendedor. Pode representar a mesma entidade que o componente anterior.

- *Data Center*: Este componente serve de intermediário entre *payment processors*. Podem ser necessários dois *data centers* numa transacção se a transacção em causa for internacional.

Uma transacção SEMOPS pode ser resumida nas seguintes fases, representadas na Figura 4:

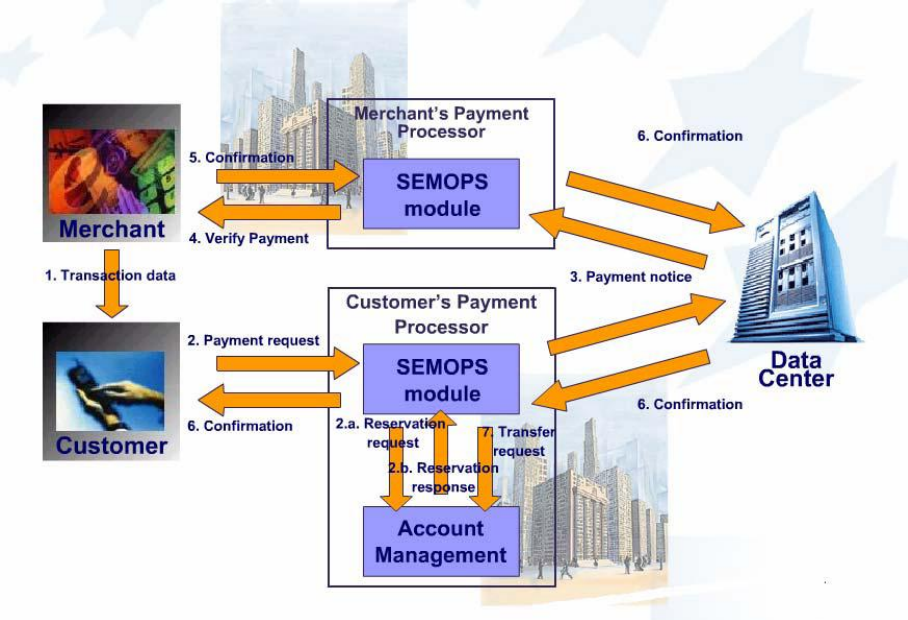


Figura 4. Arquitectura de uma transacção no projecto SEMOPS

- Passo 1. O vendedor entrega ao cliente a informação necessária para se processar o pagamento e que identifica o vendedor e a transacção.
- Passo 2. O cliente autoriza o pagamento e o seu dispositivo envia um pedido de pagamento para o CPP.
- Passo 3. O CPP verifica o saldo do cliente e reserva a quantia correspondente à transacção. De seguida comunica com o *data center*, para que este informe o MPP sobre o pagamento em curso.
- Passo 4. O MPP envia os dados da transacção ao vendedor.
- Passo 5. O vendedor confirma ou cancela a transacção comunicando a decisão ao MPP.
- Passo 6. A decisão do vendedor percorre a rede através do *data center* e do *CPP*, até chegar ao cliente. Se a decisão for uma confirmação, o CPP inicia a transferência de fundos, caso contrário liberta os que tinha reservado no passo 3. No momento em que o banco do vendedor receber a notificação de que a transferência foi bem-sucedida, o vendedor é avisado.

2.6.2 Extensões ao SEMOPS

Rahul M. Godbole e Alwyn R. Pais [22] apresentam uma extensão ao projecto SEMOPS com o objectivo de introduzir micro pagamentos mais baratos. Para o conseguir os autores propõem adicionar um novo papel ao *payment processor*. Este componente funcionaria como agregador de pagamentos de valor reduzido. Deste modo o sistema deixa de executar uma transferência bancária entre o cliente e o vendedor para cada transacção. Os *payment processors*, à medida que recebem pedidos de pagamentos, agrupam-nos até atingirem uma quantia elevada o suficiente para o custo da

transferência bancária deixar de ser significativo em relação ao valor da soma dos pagamentos. O conceito por detrás da solução apresentada pode ser resumido da seguinte forma:

- No momento em que o cliente decide efectuar um pagamento continua a contactar o seu CPP da mesma forma. No entanto, esta mensagem deixa de representar um pedido de pagamento. Nesta solução este passo passa a representar uma promessa do cliente, visto que o pagamento não será processado de imediato.
- Se o CPP executasse a transferência de imediato do cliente para o vendedor, o custo da transacção seria elevado. Logo, o CPP apenas informa o MPP sobre o pedido, e garante-lhe que o pagamento será regularizado pelo CPP posteriormente.
- Mais uma vez, O MPP não pode pagar a quantia em questão ao vendedor. Assim, assegura somente ao vendedor que lhe pagará o valor da transacção no futuro. Neste momento o vendedor pode dar a transacção como concluída, já que a sua realização foi-lhe garantida pelo banco ou operadora em que confia.
- À medida que os passos anteriores se repetem, o CPP mantém um registo do valor prometido por cada cliente. O MPP mantém por sua vez o valor que deve a cada vendedor. É também mantido o valor que cada CPP prometeu a cada MPP.
- Depois de um cliente efectuar um determinado número de pagamentos, a soma das transacções que efectuou atingirá o valor de um macro pagamento. Neste instante o CPP pode transferir o valor da conta do cliente para a sua, visto que a taxa deste movimento será baixa em relação ao valor da transacção.
- Do mesmo modo, num determinado momento, o MPP poderá pagar o valor que deve ao vendedor, sem que o custo da transferência seja significativo.
- Pelo mesmo raciocínio, após um conjunto de transacções que envolvam os mesmos *payment processors*, o CPP poderá regularizar a sua dívida para com o MPP.

2.6.3 Sistema Proposto por Kungpisdan et al.

Kungpisdan et al. [26][19] apresentam um sistema que serve como exemplo de um projecto que usa criptografia simétrica. O sistema adopta este tipo de criptografia com o objectivo de reduzir a computação necessária para executar o protocolo de pagamentos móveis.

Os componentes desta arquitectura são semelhantes aos do projecto SEMOPS. No desenho desta solução as entidades equivalentes ao CPP e MPP são representadas por bancos. Outra diferença significativa centra-se num componente denominado PG (*payment gateway)*. Este componente desempenha as funções do *data center* referido no SEMOPS. No entanto, é também responsável por interagir com o vendedor, tarefa que no SEMOPS está atribuída ao MPP. Neste sistema o cliente partilha um segredo com o CPP, enquanto o vendedor partilha um segredo com o PG. Existe também um segredo partilhado entre o cliente e cada vendedor com quem interage. A chave secreta correspondente a este último é negociada através de um protocolo AKE (*Authenticated Key Exchange*), sempre que o cliente interage com um vendedor pela primeira vez.

No momento da transacção, os passos deste protocolo são semelhantes ao SEMOPS, contudo, o protocolo difere no caminho percorrido pela transacção, como se verifica nos seguintes passos (Figura 5):

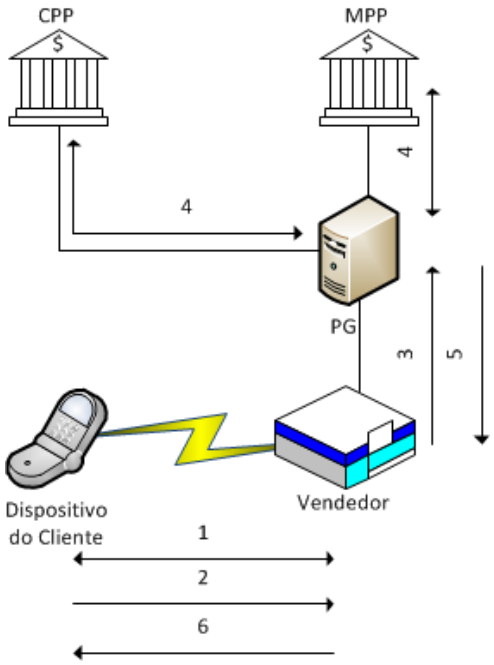


Figura 5. Arquitectura de uma transacção no sistema proposto por Kungpisdan et al.

- Passos 1 e 2: Cliente e vendedor trocam a informação necessária para se processar o pagamento, nomeadamente informação que identifica a transacção, o cliente e o vendedor. A última mensagem enviada pelo cliente ao vendedor representa o pedido de pagamento. O cliente aplica o segredo partilhado com o vendedor aos dados da transacção. O objecto resultante serve, na perspectiva do vendedor, como prova de que o cliente autoriza a transacção. O cliente repete também o processo com o segredo partilhado com o CPP, para que este possa, na presença desta mensagem, verificar que o pagamento teve o consentimento do cliente.
- Passo 3: O vendedor comunica a transacção ao PG. Na mensagem inclui a prova do pagamento a ser verificada pelo CPP. Adiciona também à mensagem um objecto que prova ao PG que o vendedor aprova a transacção, criado da forma análoga à descrita no passo anterior.
- Passo 4: O PG avisa o CPP e o MPP sobre transacção. Estes aprovam ou rejeitam o pagamento e comunicam a decisão ao PG.
- Passos 5 e 6: O PG informa o vendedor sobre o desfecho da transacção, que por sua vez avisa o cliente.

Este protocolo implica que o cliente comunica directamente apenas com o vendedor, facto cujas vantagens foram referidas anteriormente. Implica também que o conjunto dos objectos fornecidos pelo cliente ao vendedor e ao CPP, gerados a partir dos respectivos segredos, serve como prova de que o cliente aprovou a transacção. Isto porque o vendedor ou o CPP poderiam ter gerado um dos objectos, mas a única entidade que poderia, sem o auxílio de outra, ter criado ambos os objectos é o cliente.

2.6.4 mFerio

O mFerio [23] é um sistema baseado em *tokens* criado com o objectivo de substituir pagamentos em dinheiro. Este projecto suporta transferências *offline* dos tipos P2P e POS através da utilização de NFC para a comunicação de curto alcance. É adoptada uma solução criptográfica mista. É utilizada criptografia assimétrica como suporte das técnicas de assinaturas e certificados digitais, enquanto para a comunicação entre os intervenientes na transacção são usadas chaves simétricas. O sistema não especifica a comunicação com entidades centrais, apenas entre o cliente e o vendedor, ou entre o emissor e o receptor no caso de uma transacção P2P.

O serviço usa um protocolo de dois toques, isto é, os intervenientes aproximam os dispositivos móveis uma vez para, por NFC, se iniciar a transacção, aproximando-os de novo para confirmar o pagamento. O protocolo inclui também vários pontos de autenticação do utilizador perante o dispositivo. Os passos para concretizar uma transacção P2P podem ser resumidos da forma descrita de seguida (Figura 6):

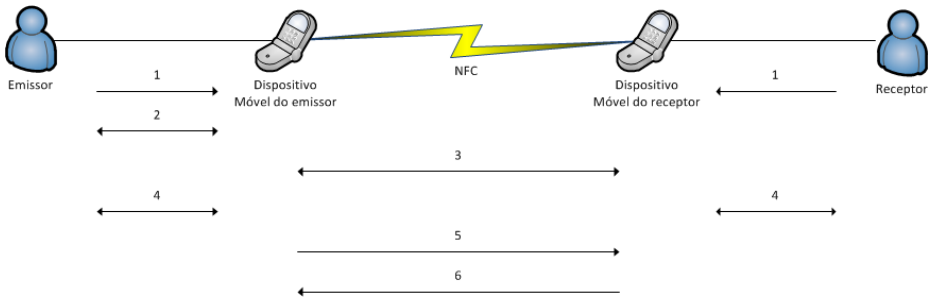


Figura 6. Arquitectura de uma transacção no sistema mFerio

- Passo 1: Ao iniciarem a aplicação ambos os intervenientes se autenticam perante os respectivos dispositivos pela primeira vez.
- Passos 2: O emissor preenche a informação relativa ao pagamento. O emissor confirma os dados introduzidos e autentica-se novamente.
- Passo 3: Os intervenientes aproximam os dispositivos, dando início à primeira fase do protocolo de dois toques. Os dispositivos trocam informação que os identifica, através de certificados digitais. Estabelecem também uma chave partilhada para o resto da transacção.
- Passo 4: O dispositivo do emissor verifica o saldo do emissor, desconta o valor do pagamento no saldo, prepara a mensagem de pagamento e assina-a. A informação adquirida no passo anterior é mostrada ao emissor, em conjunto com o resto dos dados relativos ao pagamento. O dispositivo do receptor apresenta-lhe a informação sobre a transacção. Ambos os intervenientes confirmam a transacção e repetem a autenticação perante os respectivos dispositivos.
- Passos 5 e 6: Os intervenientes aproximam os dispositivos pela segunda vez, completando o protocolo de dois toques do sistema. A transferência entre os dispositivos é então efectuada. O dispositivo do emissor do pagamento envia a mensagem preparada no passo anterior. O dispositivo do receptor verifica a assinatura, incrementa o saldo do receptor, assina um recibo da transacção e envia-o para o emissor.

2.6.5 Sistema Proposto por Zhang et al.

O sistema proposto por Zhang et al. [47] é outro exemplo de um sistema baseado em *tokens*. O sistema consegue garantir o anonimato dos pagamentos, excepto em situações em que o cliente use o mesmo *token* em vários pagamentos. Os *tokens* são também divisíveis, ou seja, o cliente pode adquirir do banco um *token* de uma quantia, e dividi-lo em *tokens* de valor mais reduzido. Este projecto foca-se na especificação do protocolo, deixando em aberto questões como as tecnologias de comunicação que deverão ser usadas. Este protocolo introduz um componente denominado *anonymous provider agent*, representado por um *smartcard* inserido no dispositivo móvel do cliente. Este componente participa na tarefa de tornar os *tokens* anónimos.

Para atingir estes objectivos, o sistema define os seguintes objectos a serem usados no protocolo:

- c. *token* que representa dinheiro no sistema.
- c'. Objecto gerado pelo cliente de modo a tornar c anónimo.
- $Cert_c$. Objecto gerado pelo banco, que expressa o seu aval na geração de c. O banco mantém uma associação entre c, $Cert_c$ e a identidade do cliente.
- $Cert_{c'}$. Objecto gerado pelo *smartcard* do cliente a partir de c' e $Cert_c$, que apesar garantir o aval do banco na criação de c', não pode ser associado directamente com o cliente, com c, ou com $Cert_c$.
- $Claim_{c'}$. Objecto gerado pelo cliente com o intuito de assegurar ao vendedor que é o autor de c', em vez de estar apenas a reencaminhar um c' recebido de outro cliente anteriormente. Se o cliente gerar dois Claims para o mesmo c', o banco tem a capacidade de detectar esta equivalência e associar este par à identidade do cliente.

As propriedades destes objectos são baseadas em técnicas como *Schnorr's undeniable signature* [48] e *blind-signature* [49]. Estas técnicas são descritas resumidamente de seguida. Os detalhes da sua implementação estão presentes em [47] e não são abordados neste documento.

Ambas as técnicas se assemelham às tradicionais técnicas de assinatura digital. No caso da *Schnorr's undeniable signature,* a diferença revela-se no acto da verificação da assinatura. Em contraste com a técnica tradicional, a verificação não pode ser efectuada localmente, através apenas da chave pública do suposto autor da mensagem. Nesta técnica, a entidade verificadora executa um protocolo de desafio-resposta com um suspeito da autoria da mensagem. Com este protocolo, o suspeito prova se é ou não o autor da mensagem. Assim, uma assinatura só pode ser verificada com o consentimento do autor da mesma.

Na técnica *blind-signature*, o autor da mensagem pretende que esta seja assinada por uma entidade. No entanto, o autor não deseja que o conteúdo da mensagem possa ser lido pelo assinante. Para o conseguir, o autor disfarça a mensagem, de modo a que o seu conteúdo seja imperceptível. De seguida, entrega esta mensagem disfarçada à entidade que a deve assinar. Esta entidade assina a mensagem ser poder ler o seu conteúdo e devolve-a ao autor. Nesta técnica, o autor consegue então retirar o disfarce da mensagem, mantendo a validade da assinatura.

O protocolo desta solução pode ser descrito em quatro fases, três das quais correspondem às fases genéricas de um sistema baseado em *tokens*. As fases do protocolo e respectivos passos são descritos de seguida (Figura 7).

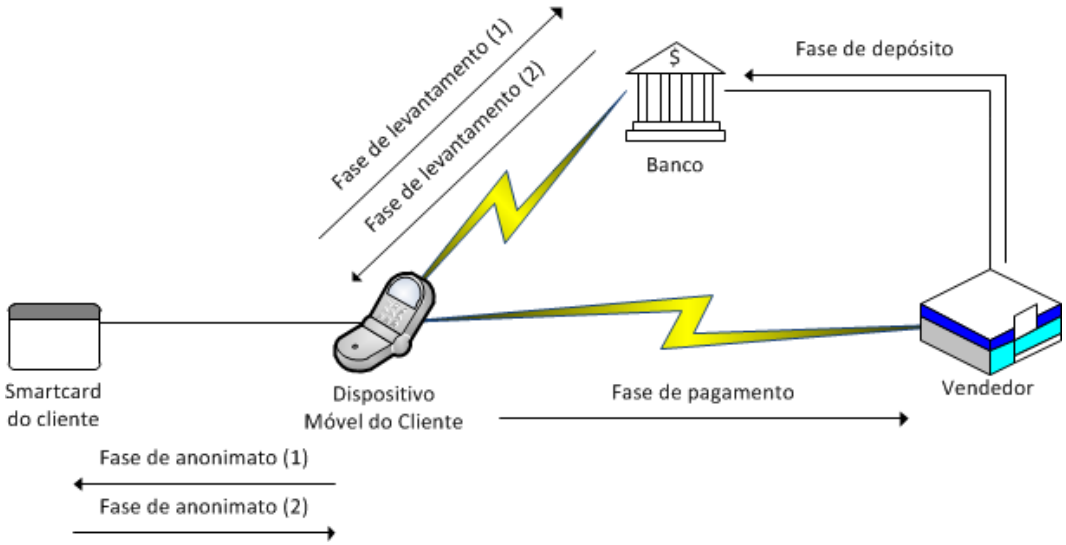


Figura 7. Arquitectura de uma transacção no sistema proposto por Zhang et al.

- Fase de levantamento (passo 1). O cliente gera um *token* c e informa o banco.
- Fase de levantamento (passo 2). Depois de verificar a validade de c, o banco decrementa o saldo do cliente, cria $Cert_c$ entrega-o ao cliente.
- Fase do anonimato (passos 1). O dispositivo do cliente gera c' e envia-o ao seu *smartcard*, em conjunto com c, $Cert_c$ e informação que lhe permite verificar a equivalência entre c e c'.
- Fase do anonimato (passo 2). O *smartcard* do cliente verifica a validade de $Cert_c$ e a equivalência entre c e c'. Se estas condições se verificarem imite $Cert_{c'}$ e entrega-o à aplicação do sistema de pagamento.
- Fase de pagamento. O cliente cria $Claim_{c'}$ e envia ao vendedor c', $Cert_{c'}$ e $Claim_{c'}$. O vendedor verifica $Cert_{c'}$ para ter a garantia do aval do banco na criação de c'. Verifica também $Claim_{c'}$ com o objectivo de garantir que c' foi gerado pelo cliente.
- Fase de depósito. O vendedor entrega ao banco os dados de cada *token* que recebeu, nomeadamente c', $Cert_{c'}$ e $Claim_{c'}$ para cada pagamento. O banco verifica a validade os dados e incrementa o valor da conta do vendedor. Entre as verificações está a detecção de reutilização ilegal de c'.

2.6.6 Sistema Proposto por Hou e Tan

A solução introduzida por Hou e Tan [21] é um exemplo de um sistema *account-based* apoiado em criptografia assimétrica, que consegue garantir anonimato e pagamentos *offline*. Para o conseguir utiliza uma técnica criptográfica denominada *group-signature* [50]. Tal como num esquema normal de criptografia assimétrica, cada entidade possui uma chave privada. A particularidade desta técnica reside na existência de uma chave pública comum a um grupo. Esta chave permite verificar assinaturas processadas por qualquer chave privada pertencente ao grupo. Não é possível todavia,

para um elemento normal do grupo, enquanto verifica a assinatura, determinar qual dos elementos do grupo é o autor da mensagem. Esta capacidade está reservada apenas para uma entidade, denominada gestor do grupo. Um componente com o nome de *Clearing House* desempenha este papel, fazendo a correspondência entre assinaturas e contas de clientes. A função de membro do grupo é cumprida pelos clientes.

O protocolo deste sistema pode ser representado pelas seguintes fases, e respectivos passos (Figura 8):

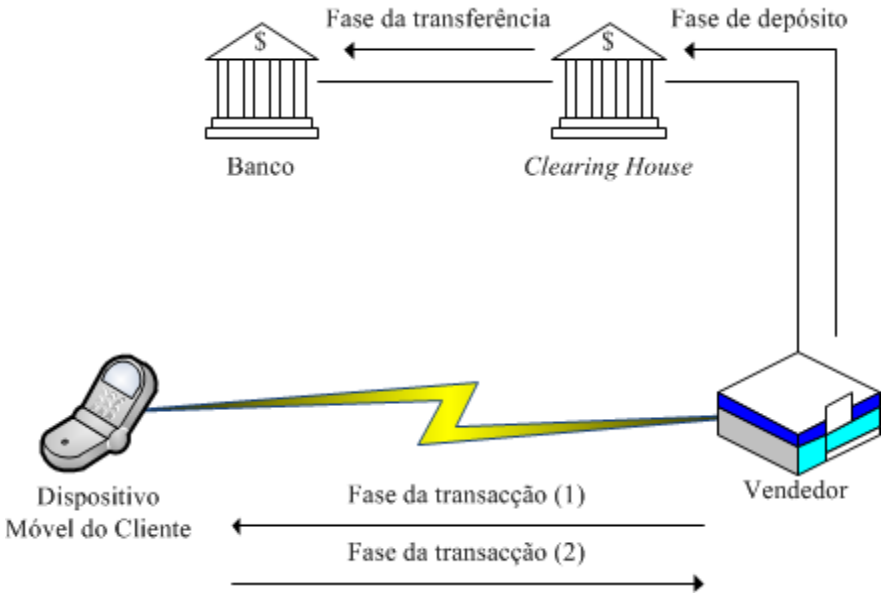


Figura 8 Arquitectura de uma transacção no sistema proposto por Hou e Tan

- Fase da transacção (passo 1). O vendedor compila a informação necessária para executar a transacção, como a identificação da mesma e do vendedor. De seguida envia estes dados ao cliente.
- Fase da transacção (passo 2). O cliente assina a transacção e devolve-a ao vendedor, demonstrando que autoriza o pagamento.
- Fase da transacção (passo 3). O vendedor verifica a assinatura. Apesar de não conseguir determinar a identidade do cliente, fica com a certeza de que a transacção foi aprovada por um cliente autorizado do sistema.
- Fase do depósito. O vendedor entrega à *Clearing House* periodicamente as mensagens de transacções acumuladas. A *Clearing House* volta a verificar as mensagens.
- Fase da transferência. A *Clearing House* indica periodicamente ao banco as transferências que devem se executadas, derivadas das transacções que acumulou. Mesmo depois desta fase as mensagens são mantidas. Em caso de uma disputa, a informação presente neste componente serve como prova dos pagamentos efectuados.

2.6.7 fairCASH

O fairCASH [24] é um sistema *token-based* e pré-pago que permite pagamentos à distância e pagamentos de proximidade *offline*. Os *tokens* deste sistema são transferíveis, isto é, podem ser

transaccionados entre clientes várias vezes antes de serem depositados. O sistema permite anonimato inquebrável dos clientes, evitando que a identificação da aplicação de pagamento seja associada com o cliente. O cliente não fornece os seus dados ao sistema através de qualquer tipo de registo. Cada entidade é identificada pelo seu certificado digital.

Os passos definidos pelo sistema, representados na Figura 9, são resumidos de seguida.

- Fase de levantamento: Para carregar a aplicação com *tokens*, o cliente efectua uma transferência bancária da quantia desejada para o sistema. Dispositivo descarrega posteriormente os *tokens* de uma das entidade centrais denominadas eMints. Para garantir a autenticidade dos *tokens*, estes são assinados pela entidade que os emite.
- Fase de pagamento: O emissor do pagamento entrega os *tokens* necessários ao receptor, assim como um 'recibo' assinado da transacção. Como foi referido, este passo pode ser repetido várias vezes. No entanto, o sistema refere um número máximo de vezes que um *token* pode ser utilizado. Para reduzir o risco de duplicação de *tokens*, cada cliente guarda um registo dos *tokens* recebidos que, com a sua permissão, é usado pelo sistema para calcular a origem de uma duplicação de *tokens*.
- Fase de depósito: Um receptor devolve os *tokens* a uma eMint. O sistema entrega posteriormente a quantia correspondente ao receptor.

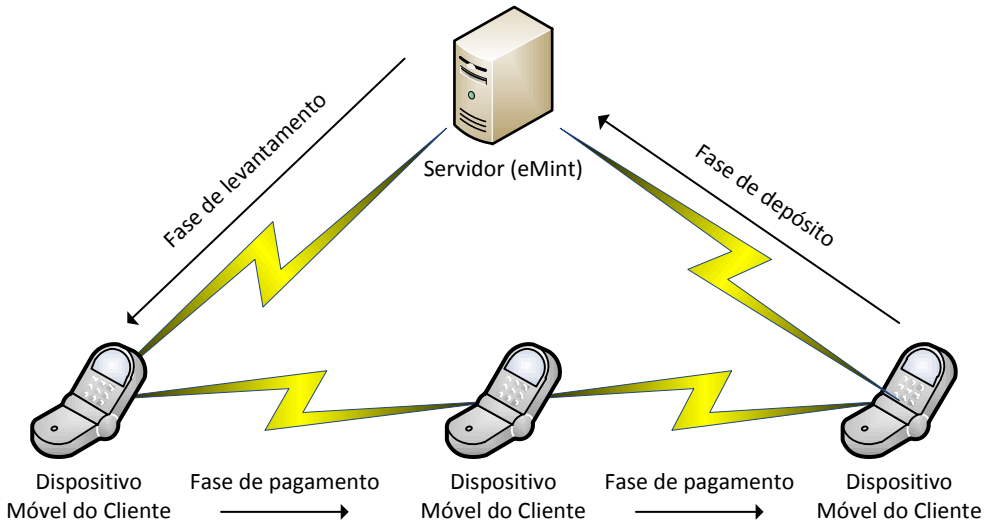


Figura 9. Arquitectura do sistema fairCASH.

2.6.8 Sistema proposto por Hassinen et al.

O sistema proposto por Hassinen et al. em [7] tira partido do projecto FINE-ID, uma PKI (*Public Key Infrastructure*) implementada na Finlândia a nível nacional. Esta iniciativa atribui um par de chaves e um certificado digital a cada cidadão, o que simplifica a implementação de sistemas que usem criptografia assimétrica, como é o caso da solução proposta por Hassinen et al.

Esta solução propõe dois protocolos de pagamento. Um dos protocolos permite pagamentos POS virtual, enquanto o outro protocolo suporta pagamentos POS *online*, com foco nos pagamentos P2M.

Os pagamentos de proximidade são suportados pela tecnologia Bluetooth. Ambos os protocolos são *account-based*.

Os passos que constituem um pagamento POS Virtual, ilustrados na Figura 10, são descritos de seguida:

- Pedido de serviço (passo 1): O dispositivo do cliente solicita ao dispositivo do vendedor a lista dos produtos disponíveis. O pedido pode conter informação que restrinja esta lista.
- Informação de serviço (passo 2): O dispositivo do vendedor envia ao dispositivo do cliente o seu certificado, assim como um lista com as descrições e preços dos produtos disponíveis.
- Selecção de produto (passo 3): O cliente comunica ao seu dispositivo qual o produto que deseja pagar. Este dispositivo envia esta selecção ao dispositivo do vendedor. A selecção enviada é assinada com a chave privada do cliente antes de ser enviada.
- Pedido de pagamento (passo 4): O dispositivo do vendedor envia ao dispositivo do banco os detalhes do pagamento. Esta informação é assinada com a chave privada do vendedor e cifrada com a chave pública do banco antes de ser enviada. Esta mensagem contém informação assinada pelo cliente no passo anterior.
- Confirmação de pagamento (passo 5): Depois de processar a transferência, o banco envia uma mensagem de confirmação do pagamento ao dispositivo do vendedor. Esta mensagem é assinada com a chave privada do banco antes de ser enviada.
- Entrega de produto (passo 6): O dispositivo do vendedor encaminha para o dispositivo do cliente a confirmação recebida do dispositivo do banco. O vendedor entrega o produto ao cliente.

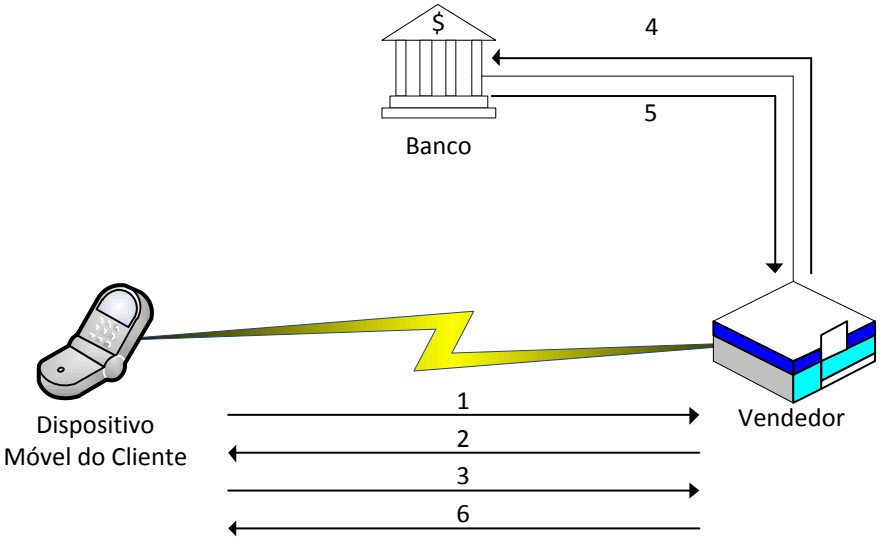


Figura 10. Passos de um pagamento POS Virtual no sistema proposto por Hassinen et al.

Os passos que constituem um pagamento POS, ilustrados na Figura 11 são descritos de seguida.

- Pré-selecção de produto (passo 1, opcional): o cliente informa o vendedor do produto que deseja.
- Oferta de serviço (passo 2): Depois de estabelecida uma ligação Bluetooth entre os dispositivos, o dispositivo do vendedor envia ao dispositivo do cliente o certificado do vendedor. Envia também a

lista dos produtos disponíveis e os respectivos preços. Se o cliente tiver seleccionado um produto através do passo 1, a lista terá apenas o produto seleccionado.

- Selecção de produto (passo 3): Se o passo 1 não tiver sido executado, o dispositivo do cliente apresenta-lhe a lista de produtos, para que o cliente seleccione o produto desejado. O dispositivo do cliente envia ao dispositivo do vendedor um mensagem composta por três partes: o certificado do cliente, a selecção feita pelo cliente cifrada com a chave pública do vendedor, e uma assinatura da selecção.
- Pedido de pagamento (passo 4): o dispositivo do vendedor envia ao dispositivo do cliente informação que identifica o vendedor e a transacção. Esta informação é assinada com a chave privada do vendedor antes de ser enviada.
- Pedido de transferência (passo 5): o dispositivo do cliente envia ao banco informação que identifica o cliente, o vendedor e a transacção. A esta informação junta a assinatura produzida pelo dispositivo do vendedor no passo anterior. Esta mensagem é assinada com a chave privada do cliente antes de ser enviada.
- Processamento de pagamento (passo 6): Para verificar a mensagem recebida, o banco obtém os certificados do cliente e do vendedor a partir do directório FINEID. Depois de verificada a mensagem recebida, o banco executa a transferência pedida. Se a transferência for bem-sucedida, o banco envia uma mensagem de confirmação ao dispositivo do cliente. Esta mensagem é assinada com a chave privada do banco e serve de prova da transacção.
- Prova de pagamento (passo 7): O dispositivo do cliente encaminha a mensagem para o dispositivo do vendedor, para que este a possa verificar. Depois de garantida a correcção da mensagem, o vendedor pode dar o pagamento como concluído e entregar o produto ao cliente.

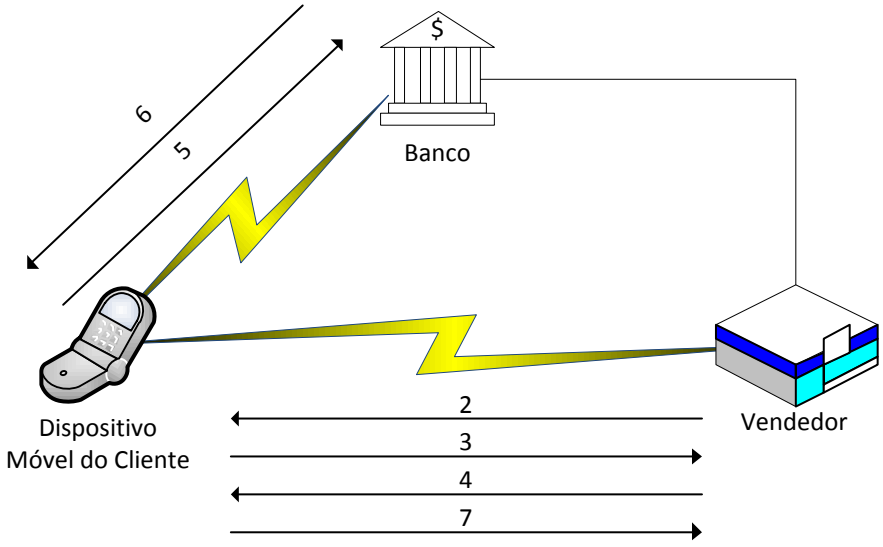


Figura 11. Passos de um pagamento POS no sistema proposto por Hassinen et al.

2.6.9 Sistemas de Pagamentos Móveis em Portugal

Em Portugal, a SIBS [51] disponibiliza, em parceria com as operadoras Vodafone, TMN e Optimus, o serviço MB PHONE [52]. Este serviço, disponível desde 1996, tem como objectivo oferecer as

mesmas funcionalidades presentes nos ATMs (*Automated Teller Machine*), como consulta de saldo, pagamento de serviços ou transferências entre contas. A aplicação MB PHONE pode funcionar por chamada de voz, SMS ou por WAP. Em comparação com o serviço nos tradicionais ATMs, cada operação tem um custo adicional associado à comunicação, que pode variar de operadora para operadora.

A Movensis, em colaboração com a Caixa Geral de Depósitos, desenvolveu também um sistema de pagamentos móveis, o CGD *Mobile Payments* [53]. Este sistema disponibiliza o serviço sobre SMS ou através de códigos de barras 2D. O CGD *Mobile Payments* foi concebido para ser utilizado em POS e P2M. Está também prevista a sua utilização em bilhética. Neste tipo de situações, o bilhete é armazenado no telemóvel em SMS ou código de barras 2D.

2.6.10 Comparação entre os Sistemas Abordados

A Tabela 1 oferece uma comparação entre os sistemas descritos anteriormente. Esta comparação é resumida imediatamente após a apresentação da tabela.

Tabela 1. Comparação dos sistemas abordados em função das suas funcionalidades

	SEMOPS	Kungpisdan et al.	mFerio	Zhang et al.	Hou e Tan	fairCASH	Hassinen et al
Valor Monetário das Transacções	Todos	Não especificado	Todos	Todos	Não especificado	Não especificado	Não especificado
Tipo de Interacção	Todos	POS	POS e P2P	Não especificado	Não especificado	Todos	POS e POS Virtual
Momento do Pagamento	Todos	Não especificado	Não especificado	Pré-pago	Pós-pago	Pré-pago	Tempo real
Tipo de Transacção	*Account-based*	*Account-based*	*Token-based*	*Token-based*	*Account-based*	*Token-based*	*Account-based*
Necessidade de Intermediários	*online*	*online*	*offline*	*offline*	*offline*	*offline*	*online*
Tipo de Criptografia	Mista	Simétrica	Mista	Assimétrica	Assimétrica	Assimétrica	Assimétrica
Anonimato	Parcial[2]	Não suportado	Apenas em parte do sistema[3]	Sim.	Parcial [4]	Sim	Não suportado
Quebra de Anonimato	Não		Em caso de disputa	Em caso de uso indevido	Mediante autorização de um juiz ou do cliente	Não	

Na primeira coluna confirmam-se as características já referidas do sistema SEMOPS. Este é o sistema mais universal e versátil, como se verifica pelas três primeiras linhas. Todos os seus atributos estão especificados, o que reflecte o facto de o sistema estar definido de forma mais completa que os restantes. As suas principais desvantagens estão representadas nas linhas "Necessidade de Intermediários" e "Anonimato".

O sistema representado na segunda coluna é o único baseado em criptografia simétrica, com as consequências referidas na secção 2.3.6 deste documento. Dos cinco sistemas apresentados este é, a par do proposto por Hassinen et al, o que se encontra pior classificado relativamente ao atributo "Anonimato".

O mFerio é o primeiro sistema baseado em *tokens* desta lista. Tem como principal vantagem permitir transacções P2P *offline*. Em relação aos restantes sistemas, este é o que está definido de forma mais incompleta. O mFerio não especifica, por exemplo, a interacção com entidades que não sejam os dois intervenientes da transacção.

[2] Apenas do cliente em relação ao vendedor

[3] Informação da transacção restringida aos intervenientes (emissor e receptor).

[4] Entre todas as entidades, com excepção da *Clearing House*

Apesar dos dois sistemas seguintes especificarem parte da interacção com a entidade central da arquitectura, falham na especificação dos tipos de interacção que suportam. Na quarta coluna pode verificar-se que o sistema proposto por Zhang et al. é o único que permite, simultaneamente, anonimato em todo sistema e possibilidade de quebra do mesmo. A quinta coluna apresenta o único sistema *Account-based* desta tabela que permite transacções *offline*.

Da penúltima coluna destaca-se o anonimato mais completo da tabela, que pode ser observado nas duas últimas linhas.

Na coluna final apresenta-se um exemplo de um protocolo com criptografia assimétrica sem possibilidade de anonimato. Com excepção do SEMOPS, este é o único sistema com pagamentos em tempo real.

3 Arquitectura

O sistema de suporte a pagamentos móveis ePaga pretende comportar todos os tipos de pagamentos referidos, assim como várias tecnologias de comunicação em modo de infra-estrutura.

Para atingir estes objectivos, foi efectuada uma análise de protocolos de pagamento existentes. A fase seguinte passou por definir uma estrutura que, aproveitando o que estes protocolos têm em comum, permita criar um sistema que os suporte. Com base nesta análise foram tomadas decisões a três níveis: tecnologias de comunicação, operações suportadas e componentes que compõem o sistema.

Em relação a tecnologias de comunicação de curto alcance, foi escolhida a tecnologia NFC, por ser a única que não limita o sistema a nível de segurança e usabilidade. Para além destas razões, o NFC está a ser adoptado por organismos internacionais na área financeira e de telefonia móvel, como é o caso do EPC, da GSMA e da GlobalPlatform. A escolha das tecnologias de comunicação de longo alcance foi deixada em aberto, de modo a não reduzir a compatibilidade do sistema.

Para criar uma aplicação que suporte os vários protocolos de pagamento nas diversas interacções que estes acarretam, foi necessário definir quais os componentes cuja comunicação e implementação tem de ser prevista pelo sistema. Foram usados os protocolos estudados para responder às perguntas: “Quais dos actores têm de estar representados no sistema?”; “Que actores têm interacções suficientemente semelhantes para poderem ser agrupados?”. Estes componentes são definidos com o maior grau de liberdade possível, de modo a não restringir o número de protocolos suportados pela solução proposta.

De seguida, foi necessário especificar quais as operações que o sistema oferece aos seus utilizadores. Apenas com esta informação é possível especificar a interface que o sistema deve apresentar ao utilizador e aos protocolos que suporta. E também necessária para prever a comunicação entre os componentes previamente referidos. A definição das operações responde às perguntas: “Que acções têm de ser previstas pelo sistema para permitir o normal funcionamento dos protocolos que suporta?”; “Quais os passos que devem ser especificados pelo sistema e quais devem ser deixados à responsabilidade do protocolo?”

Nas secções seguintes deste capítulo são descritos os componentes reconhecidos pelo sistema, a arquitectura da aplicação utilizada pelo cliente, as operações definidas pelo sistema e o desenvolvimento de protocolos compatíveis com o sistema.

3.1 Aplicação Cliente

O sistema ePaga permite que o dispositivo do cliente contenha vários protocolos de vários sistemas de pagamentos, com características distintas. Estes protocolos convivem de forma transparente do ponto de vista do utilizador do sistema, que observa apenas uma aplicação. Para cumprir este objectivo, a aplicação do dispositivo do cliente é composta por duas partes. A parte principal da aplicação do cliente encontra-se no sistema operativo do telemóvel como Android,

Symbian ou iOS. Esta secção da aplicação é independente dos protocolos instalados. A outra parte da aplicação encontra-se no elemento seguro do dispositivo, isolada do resto do dispositivo.

A Figura 12 representa a arquitectura da aplicação do dispositivo móvel. A camada superior implementa a interacção com o utilizador, de modo a que a aplicação lhe apresente uma interface consistente. A camada imediatamente abaixo da camada de apresentação aloja os vários protocolos de pagamento. Estes protocolos implementam formas distintas de representar um sistema de pagamentos móveis. Esta é a única parte da arquitectura que varia entre sistemas de pagamento. O *middleware* do sistema visa oferecer uma camada de abstracção aos protocolos de pagamento que sejam implementados sobre ele. Este *middleware* deve também, no instante de receber ou efectuar um pagamento, escolher o protocolo da camada superior a ser usado. Os blocos da camada inferior representam funcionalidades disponibilizadas pelo dispositivo e que são utilizadas pelo *middleware*.

A camada de *middleware* está dividida nos seguintes módulos:

- Mecanismos de Execução de Operações: gerem a execução da parte das operações definidas pelo sistema que é comum entre os protocolos. Fazem parte do fragmento principal da aplicação.
- Mecanismos de Escolha de Protocolo: escolhem os protocolos contidos no Registo de Protocolos que são elegíveis para executar um pagamento. Usam a Política de Escolha de Protocolo para ordenarem estes protocolos por ordem de preferência. Fazem parte do elemento seguro.
- Registo de Protocolos: mantém a informação sobre os protocolos de pagamento instalados no dispositivo. Faz parte do elemento seguro.
- Armazenamento de Serviço: mantém a informação sobre o serviço disponibilizado pelo dispositivo. Este componente é usado apenas quando o dispositivo está a receber um pagamento. Faz parte do elemento seguro.
- Política de Escolha de Protocolo: conjunto de regras que, dado um conjunto de protocolos e informação de contexto, ordenam o conjunto de protocolos segundo um determinado critério, que deve ser configurável. Faz parte do elemento seguro.
- Comunicação Local: gere a comunicação NFC com outros dispositivos próximos. Gere também o acesso local ao elemento seguro do dispositivo. Faz parte do fragmento principal da aplicação.
- Comunicação Remota: gere a comunicação remota com base em *web services*. Faz parte do fragmento principal da aplicação.

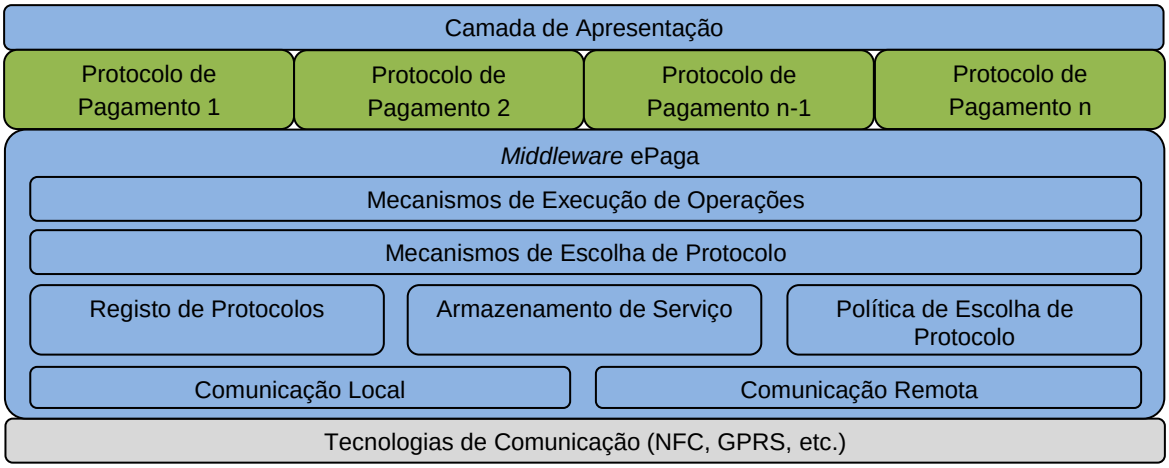


Figura 12. Arquitectura de *software* da aplicação cliente do sistema ePaga.

3.2 Componentes

O sistema ePaga define cinco entidades que interagem nas operações de pagamento: dispositivo do cliente, elemento seguro interno, elemento seguro externo, vendedor remoto e provedor de serviço remoto. Estes componentes e a forma como comunicam estão representados na Figura 13.

- Dispositivo do cliente: aloja o fragmento principal da aplicação do cliente. Em conjunto com o elemento seguro interno constitui a aplicação do cliente, cuja arquitectura é abordada na secção 3.1. Esta aplicação interage com o cliente e com os outros quatro componentes do sistema. Funciona assim como intermediário entre as entidades do sistema, mas não contém a lógica do sistema. Mantém apenas *soft-state*, logo este componente pode ser substituído por outro equivalente sem que a aplicação cliente deixe de funcionar.
- Elemento seguro interno: aloja a parte crítica da aplicação cliente, incluindo protocolos de pagamento e políticas de escolha de protocolos.
- Elemento seguro externo: constitui o receptor de uma transacção local. Pode representar o dispositivo de um vendedor ou, numa transacção P2P, o dispositivo de outro cliente.
- Vendedor remoto: representa o dispositivo do vendedor numa transacção remota. Da implementação deste componente, o sistema apenas define uma interface que tem de ser respeitada, para que a aplicação cliente possa aceder a qualquer vendedor remoto da mesma forma. Os restantes detalhes da sua arquitectura variam consoante as implementações dos vários protocolos.
- Provedor de serviço remoto: servidor que representa a entidade central do sistema. Tal como para o vendedor remoto, apenas é definida uma interface comum entre provedores.

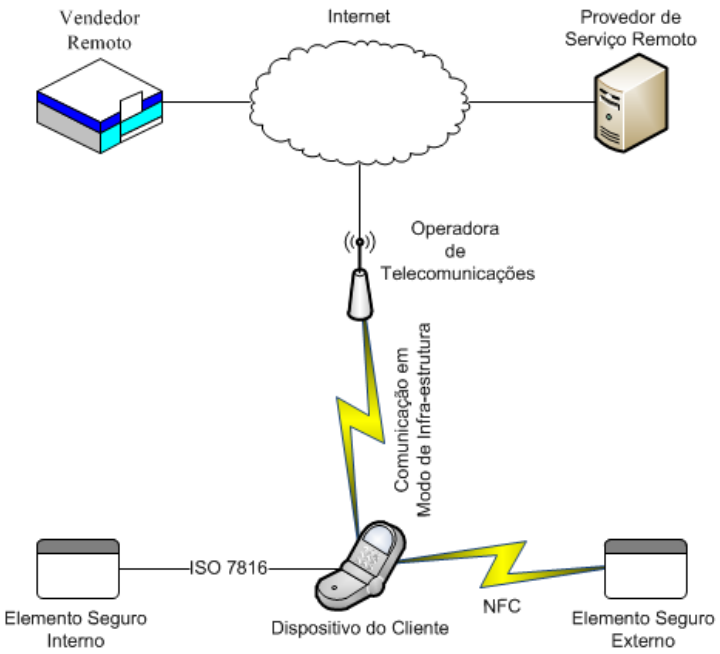


Figura 13. Diagrama de rede simplificado que ilustra os principais componentes da arquitectura do sistema ePaga.

3.3 Operações

No levantamento efectuado sobre sistemas de pagamento electrónicos, foram identificadas 4 operações que estes sistemas oferecem aos seus utilizadores. O sistema ePaga tem de comportar estas operações de modo a maximizar o número de protocolos de pagamento suportados. A arquitectura do sistema define as seguintes operações: levantamento, pagamento, verificação de saldo e depósito. A operação de pagamento divide-se ainda em pagamento local, pagamento remoto e recepção de pagamento.

Num pagamento P2P ambos os intervenientes podem ser considerados clientes. Como tal quando, a partir deste ponto, a designação cliente e vendedor se tornar ambígua, a entidade que efectua o pagamento é denominada emissor, enquanto a que recebe o pagamento é denominada receptor.

3.3.1 Levantamento

A operação de levantamento corresponde a um “carregamento” de um protocolo de pagamento. Para os protocolos que incluem uma fase levantamento, como descrito anteriormente para protocolos *token-based*, esta é a operação que o utilizador pode executar para despoletar esse levantamento.

Dependendo do protocolo, a execução desta operação pelo dispositivo móvel pode necessitar de ser complementada por acções noutro sistema. É possível, por exemplo, desenvolver um protocolo de pagamento em que o cliente, para utilizar o sistema, tem de transferir *a priori* a quantia que deseja utilizar por outros meios como multibanco. O cliente executaria de seguida a operação de levantamento no dispositivo móvel, que causaria uma sincronização do saldo no dispositivo móvel. Por outro lado, é também possível desenvolver um protocolo que, no momento da operação de levantamento do dispositivo móvel, desencadeia todas as acções necessárias para actualizar o saldo

do utilizador. Estas acções podem concretizar-se, por exemplo, numa actualização de um valor a pagar no final do mês (nos sistemas pós-pagos), ou na execução de uma transferência bancária (nos sistema em tempo real).

De seguida descreve-se o fluxo definido pelo sistema para uma operação de levantamento representado na Figura 14.

O utilizador escolhe o protocolo que deseja carregar a partir de uma lista apresentada pelo dispositivo. A aplicação móvel ePaga informa o protocolo escolhido sobre a intenção do utilizador em executar um levantamento (bloco “Pedido de Levantamento” na figura). A aplicação segue os passos definidos pelo protocolo para uma operação de levantamento (bloco “ Execução do Levantamento”).

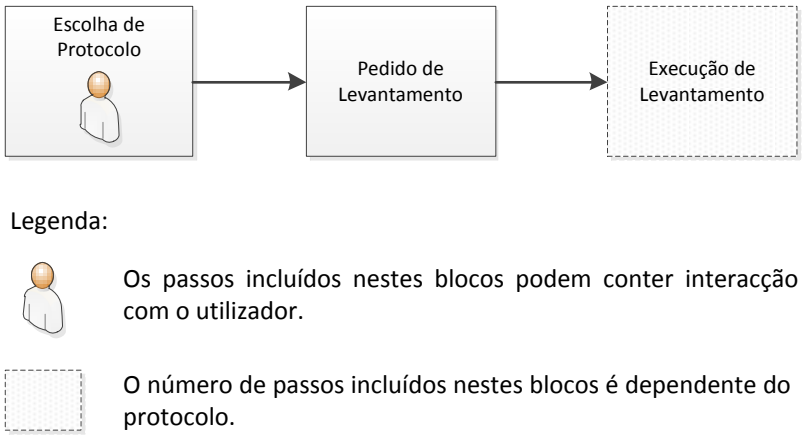


Figura 14. Fluxo da operação de levantamento.

Os passos definidos pelo sistema para cada um dos blocos referidos são descritos de seguida (Figura 15).

Escolha de protocolo (passo 1): o dispositivo do cliente pede ao seu elemento seguro a lista de protocolos instalados. O módulo Registo de Protocolos é o responsável por responder ao pedido.

Escolha de protocolo (passo 2): são mostrados ao cliente os protocolos que suportam levantamentos. O utilizador escolhe o protocolo que deseja carregar.

Pedido de levantamento: o dispositivo do cliente entrega o valor do levantamento ao protocolo escolhido. O conteúdo do pedido é independente do protocolo, mas o mesmo não se verifica para o conteúdo da resposta. A implementação do processamento do passo é dependente do protocolo.

Execução de levantamento (passos 1 a N): o dispositivo do cliente executa a operação de levantamento do protocolo escolhido. Os componentes envolvidos, as mensagens trocadas, o número de passos N e a implementação de cada passo são dependentes do protocolo.

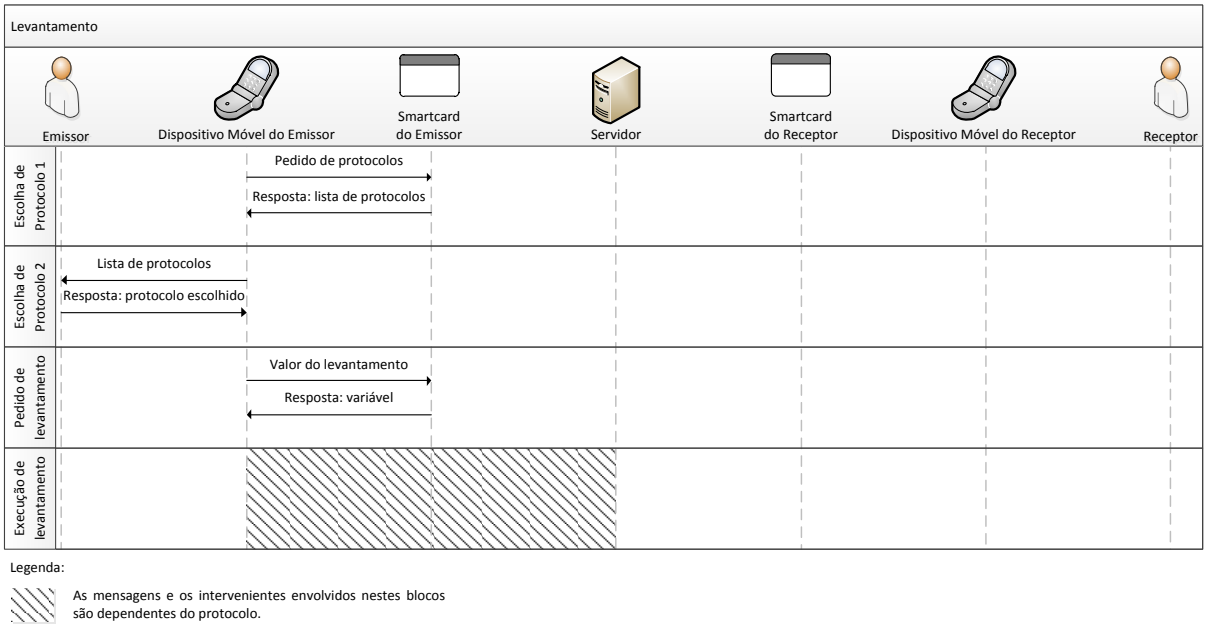


Figura 15. Passos definidos para a operação de levantamento.

3.3.2 Pagamento

A operação de pagamento constitui a acção principal de um sistema de pagamento e representa uma transferência de créditos entre duas entidades. Estes créditos podem ou não ter uma relação com uma unidade monetária. Como foi referido, as entidades envolvidas podem variar entre pagamentos. O receptor pode ser um vendedor ou um cliente. Pode estar próximo do emissor, sendo utilizado NFC para estabelecer a ligação entre o dispositivo do emissor e do receptor. Pode encontrar-se num espaço físico diferente do emissor, caso em que o dispositivo móvel faz uso da tecnologia de comunicação em modo de infra-estrutura disponível (GSM, UMTS, LTE, etc.).

De seguida descreve-se o fluxo definido pelo sistema para uma operação de pagamento representado na Figura 16.

A operação de pagamento é iniciada pelo receptor do pagamento. O receptor indica ao seu dispositivo a possibilidade de receber um pagamento (Criação de Serviço). Posteriormente, o dispositivo do emissor determina qual o pagamento que o receptor deseja receber (Pedido de Serviço). A partir da informação de pagamento recolhida na acção anterior (Pedido de Serviço), os dispositivos de emissor e receptor negoceiam o protocolo mais adequado para processar o pagamento (Pedido de Protocolos). Este protocolo é então utilizado para executar a Negociação de Pagamento. Esta acção tem como objectivo assegurar que ambos os dispositivos têm acesso a todos os parâmetros do pagamento. Com esta informação os dispositivos verificam as condições necessárias para que o resto do protocolo seja executado. Depois de os dispositivos garantirem que é possível executar o protocolo, o emissor e o receptor têm de autorizar o pagamento. Esta autorização é verificada na acção Confirmação. Depois de a transacção ser confirmada por ambos os intervenientes, é iniciada a última acção da operação de pagamento (Fase de Transferência). Nesta

fase são executados os passos necessários para que os créditos que representam o pagamento passem da posse do emissor para o receptor. Estes passos são definidos pelo protocolo de pagamento.

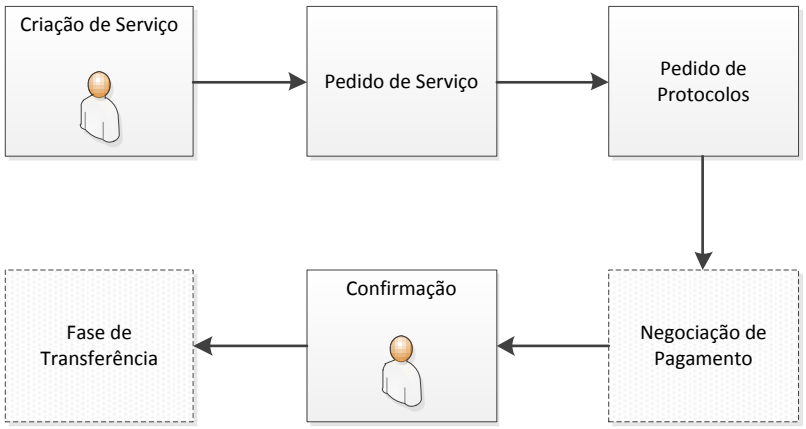


Figura 16. Fluxo da operação de pagamento.

Os passos definidos pelo sistema para cada um dos blocos referidos são descritos de seguida.

Criação de serviço (Figura 17): a transacção começa com a criação de um serviço. Este passo consiste em fornecer ao sistema a informação sobre a transacção necessária para escolher um protocolo de pagamento. Esta informação contém, no mínimo, o valor da transacção. A informação pode ser recolhida de forma manual, sendo introduzida pelo receptor no dispositivo que o representa, ou de forma automática, recolhida a partir de outras fontes como etiquetas NFC ou códigos de barras.

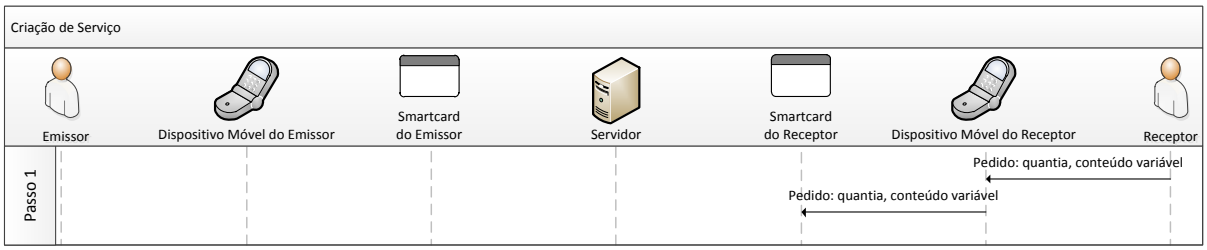


Figura 17. Passos definidos para o bloco criação de serviço.

Início de fase de negociação: se a transacção em questão se tratar de um pagamento de proximidade, o emissor e o receptor aproximam os dispositivos. É executada a fase de negociação. Nesta fase deve ser escolhido o protocolo a usar e trocada a informação necessária para os intervenientes possam tomar uma decisão em relação a transacção.

Pedido de serviço: o objectivo deste bloco é que o dispositivo do emissor obtenha a informação relativa ao serviço disponibilizado pelo dispositivo do receptor. Esta inclui, no mínimo, o valor do pagamento.

No caso de um pagamento remoto, poderá ser necessário fornecer à aplicação o endereço da entidade que faculta a informação relativa ao serviço. Esta informação enquadra-se nos típicos pagamentos POS virtual, em que o pagamento é precedido de uma interacção com uma página Web do vendedor. Como tal, esta informação deverá ser preenchida automaticamente. Nos casos em que não existe uma entidade a quem se possa pedir a informação do serviço, o emissor terá de introduzir manualmente a informação relativa à transacção, nomeadamente o valor do pagamento e a identificação do destinatário. Este caso aplica-se a transacções como transferências bancárias.

Pedido de serviço (passo 1): o endereço que aloja a oferta de serviço é introduzido na aplicação, de forma manual (Figura 19) ou automática. Este passo apenas é executado para pagamentos remotos.

Pedido de serviço (passo 2): se o passo anterior tiver sido executado e valor do endereço tenha sido deixado em branco, o emissor introduz no seu dispositivo a informação necessária para se processar a transacção, que inclui a identificação do destinatário da transacção e o valor da mesma (Figura 20). Caso contrário, o dispositivo do emissor faz um pedido de serviço ao dispositivo que representa o receptor. O pedido é enviado ao componente vendedor remoto (Figura 18 ou Figura 19) ou ao elemento seguro externo (Figura 21), dependendo do pagamento ser à distância ou de proximidade respectivamente.

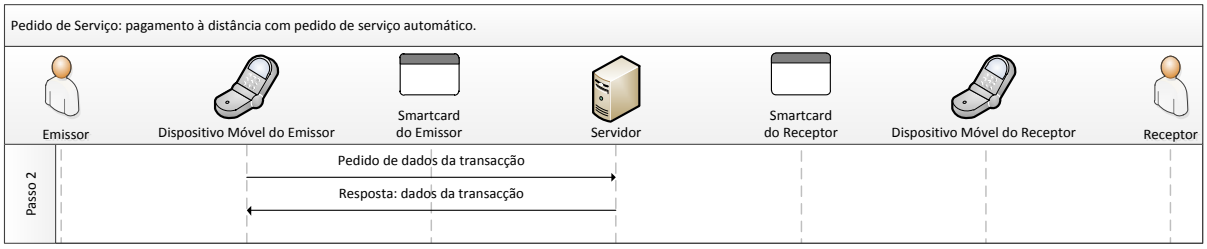


Figura 18. Passos de um pedido de serviço automático para pagamentos à distância.

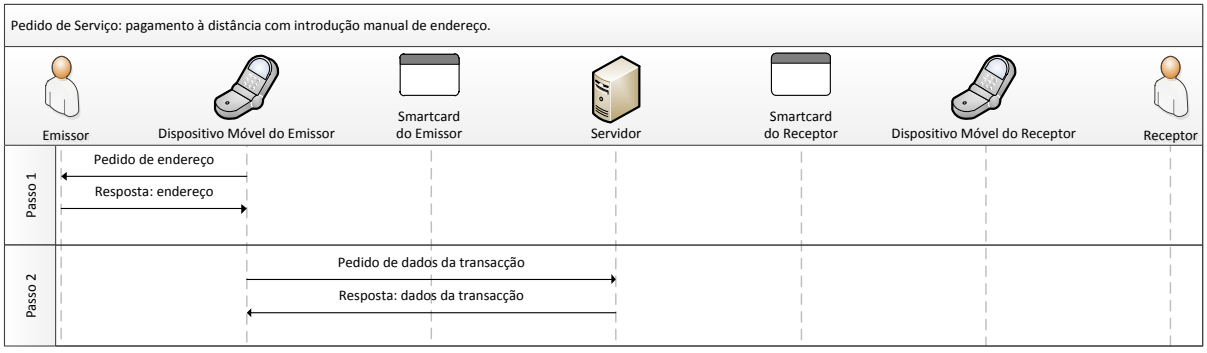


Figura 19. Passos de um pedido de serviço semiautomático para pagamentos à distância.

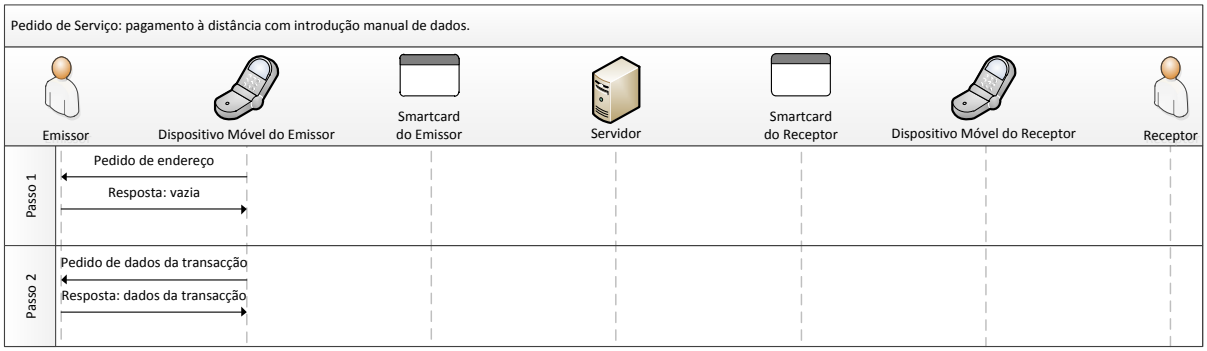


Figura 20. Passos de um pedido de serviço manual para pagamentos à distância.

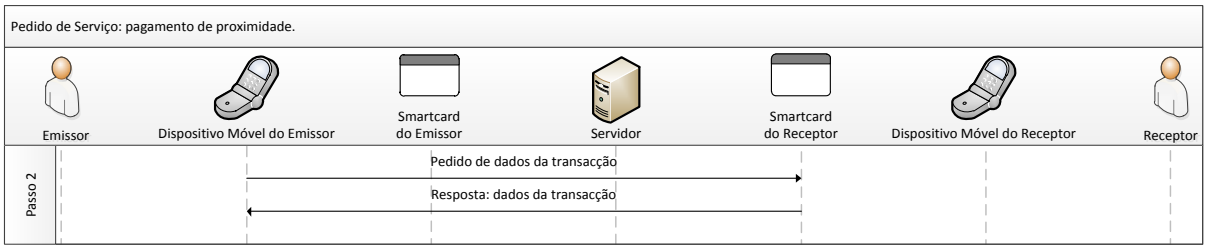


Figura 21. Passos de um pedido de serviço para pagamentos de proximidade.

Pedido de protocolos (Figura 22): o dispositivo do emissor prepara uma lista de protocolos disponíveis para executar o pagamento, ordenada por ordem de preferência.

Pedido de protocolos (passo 1): o dispositivo do emissor pede ao seu elemento seguro a lista referida. No pedido inclui informação de contexto, nomeadamente informação que categoriza o pagamento como de proximidade ou à distância, a oferta de serviço e o estado da ligação à rede de dados do dispositivo.

Pedido de protocolos (passo 2): o módulo Mecanismos de Escolha de Protocolos é o responsável por responder ao pedido. Este módulo começa por solicitar a lista de protocolos instalados no dispositivo ao módulo Registo de Protocolos.

Pedido de protocolos (passo 3): o módulo Mecanismos de Escolha de Protocolos filtra a lista recebida de modo a manter apenas os protocolos que se adequam ao tipo de pagamento. Este processo implica, se a ligação à rede não estiver disponível, descartar protocolos que apenas suportam pagamentos *online*, descartar protocolos à distância em pagamentos de proximidade ou descartar protocolos de proximidade em pagamentos à distância.

Pedido de protocolos (passo 4): o módulo Mecanismos de Escolha de Protocolos entrega ao módulo Política de Escolha de Protocolo a lista filtrada no passo anterior. Este último módulo inicializa a política configurada e utiliza-a para ordenar a lista de protocolos por ordem de preferência. O critério de escolha dos protocolos varia entre políticas. A lista ordenada é encaminhada como resposta aos pedidos até ser entregue à aplicação principal do emissor.

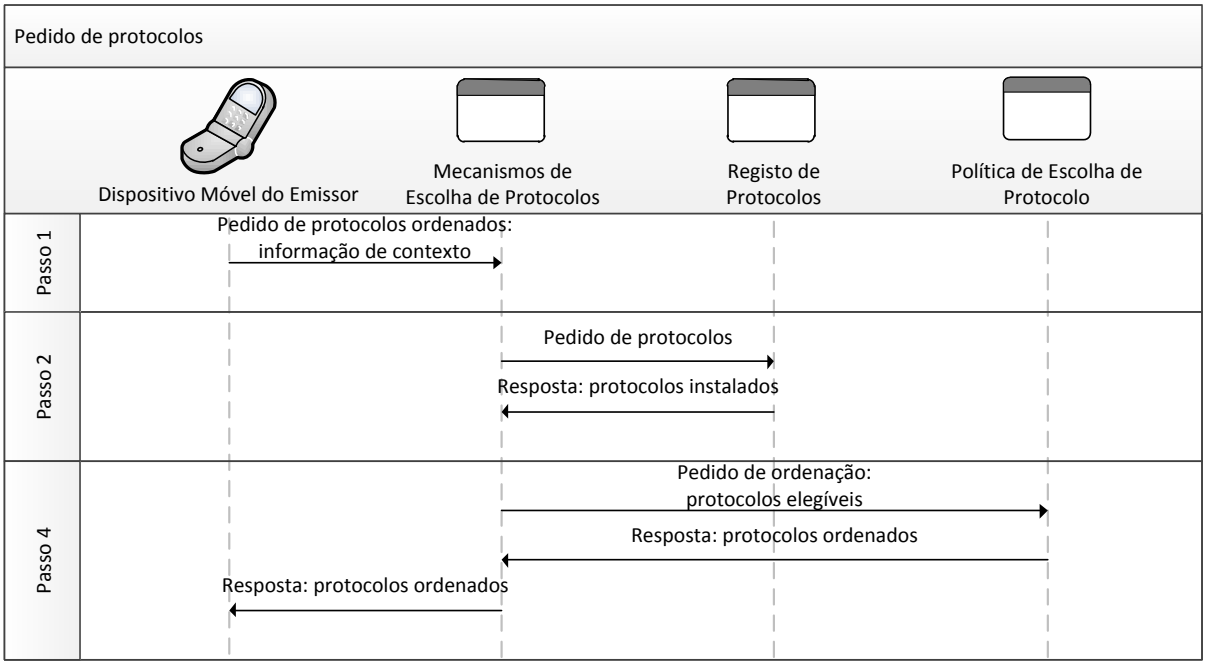


Figura 22. Passos definidos para o bloco pedido de protocolos.

Negociação de pagamento: o dispositivo do emissor tenta negociar o pagamento usando cada um dos protocolos até ser bem-sucedido. Os passos desta negociação são parcialmente dependentes do protocolo.

Negociação de pagamento (passo 1): o dispositivo do emissor efectua um pedido de negociação ao elemento seguro. No pedido inclui a data actual e a oferta de serviço adquirida no pedido de serviço. A data serve para introduzir o conceito de tempo no elemento seguro, que é necessário em alguns protocolos. Se o protocolo escolhido não lidar com datas pode ignorar este campo. A oferta de serviço, como já foi referido, varia entre protocolos e contém, no mínimo, o valor do pagamento. O processamento da mensagem e o conteúdo da resposta ao pedido são dependentes do protocolo.

Negociação de pagamento (passos 2 a N-1): O dispositivo executa os passos definidos pelo protocolo para a operação de negociação. Os componentes envolvidos, as mensagens trocadas, o número de passos N e a implementação de cada passo são dependentes do protocolo.

Negociação de pagamento (passo N): Se a transacção em causa se tratar de um pagamento de proximidade, o dispositivo do emissor notifica o elemento seguro do receptor sobre o protocolo escolhido. A informação do protocolo escolhido visa facilitar os passos seguintes, mas não é obrigatório para o sucesso da operação de pagamento. Este passo marca o final da fase de negociação.

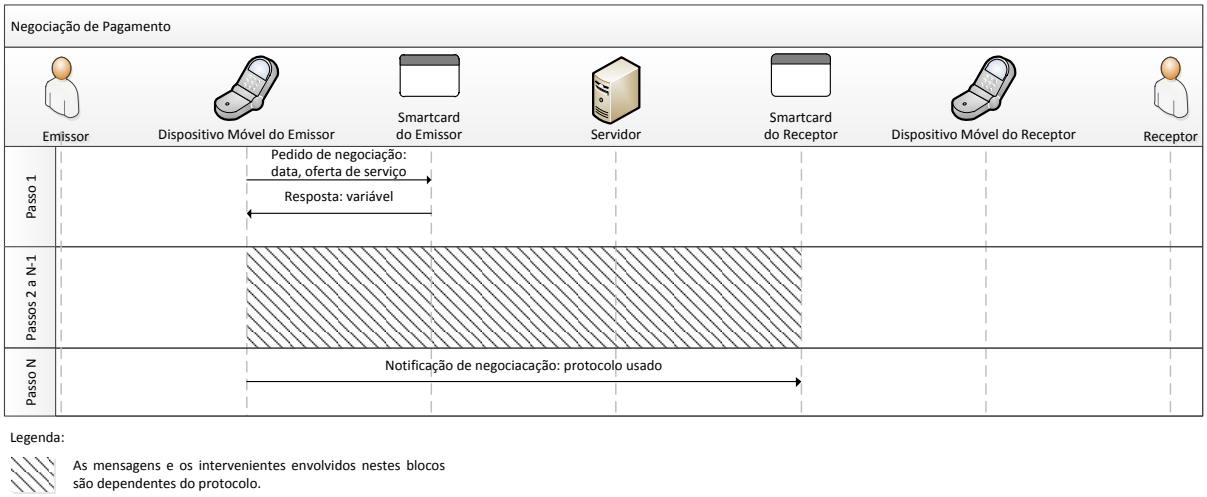


Figura 23. Passos definidos para o bloco negociação de pagamento.

Confirmação: a informação relativa a transacção é apresentada ao emissor e ao receptor, que confirmam ou cancelam a operação. Para o fazer, o emissor (Figura 24) e o receptor (Figura 25 e Figura 26) têm de se autenticar perante os respectivos dispositivos. A confirmação do receptor apenas é executada em pagamentos de proximidade.

Confirmação do emissor (passo 1): o dispositivo do emissor pede ao seu elemento seguro para preparar a transacção utilizando os dados de saída da negociação de pagamento. O elemento seguro responde-lhe com a identificação do emissor, identificação do receptor e com o valor do pagamento para que esta informação possa ser mostrada ao emissor.

Confirmação do emissor (passo 2): o emissor confirma ou cancela a transacção. Para confirmar tem de introduzir o código PIN (*Personal Identification Number*) da aplicação no seu dispositivo. O dispositivo do emissor entrega esta informação ao respectivo elemento seguro para que o PIN possa ser verificado. No caso de o emissor cancelar transacção não precisa de introduzir o PIN. Neste caso dispositivo do emissor também entrega esta informação ao respectivo elemento seguro, para que este volte ao estado inicial.

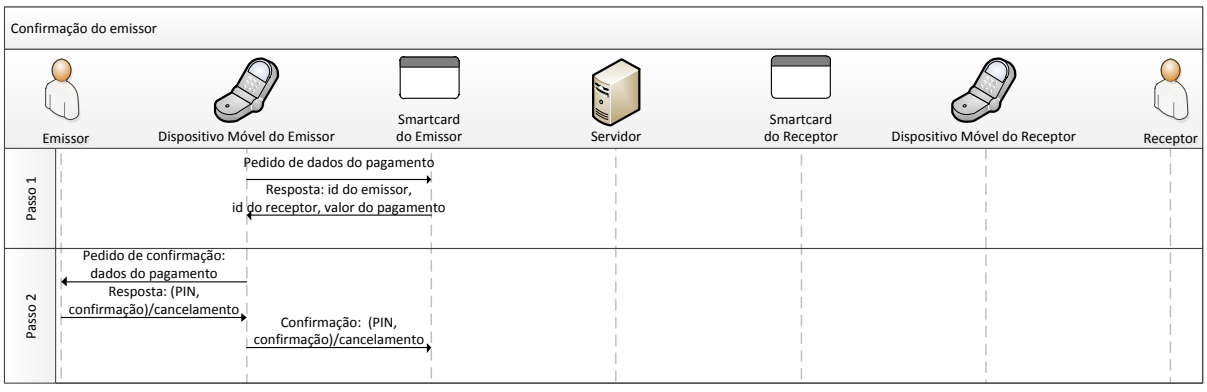


Figura 24. Passos definidos para a confirmação do emissor.

Confirmação do receptor (passo 1): o dispositivo do receptor solicita ao módulo Registo de Protocolos informação sobre o protocolo que detém a informação da transacção. Se este passo for bem-sucedido, o dispositivo salta para o passo 3.

Confirmação do receptor (passo 2): se o dispositivo do emissor, por não estar a utilizar o sistema ePaga, não tiver fornecido a informação do passo anterior ao módulo Registo de Protocolos, o dispositivo do receptor tem de inquirir todos os protocolos instalados. Para o fazer começa por pedir ao módulo Registo de Protocolos a lista de protocolos instalados.

Confirmação do receptor (passo 3): se tiver sido executado o passo 2, este passo deve ser executado para cada protocolo da lista. O dispositivo do receptor pede ao seu elemento seguro, mais concretamente ao protocolo em uso, a informação da transacção que será mostrada ao receptor. Esta informação é exactamente a mesma que é mostrada ao emissor. As mensagens trocadas são independentes do protocolo. A implementação do passo no elemento seguro é dependente do protocolo.

Confirmação do receptor (passo 4): o receptor confirma ou cancela a transacção. Para confirmar tem de introduzir o código PIN da aplicação no seu dispositivo. O dispositivo do receptor entrega esta informação ao respectivo elemento seguro para que o PIN possa ser verificado. No caso de o receptor cancelar transacção não precisa de introduzir o PIN. Neste caso dispositivo do receptor também entrega esta informação ao respectivo elemento seguro, para que este volte ao estado inicial.

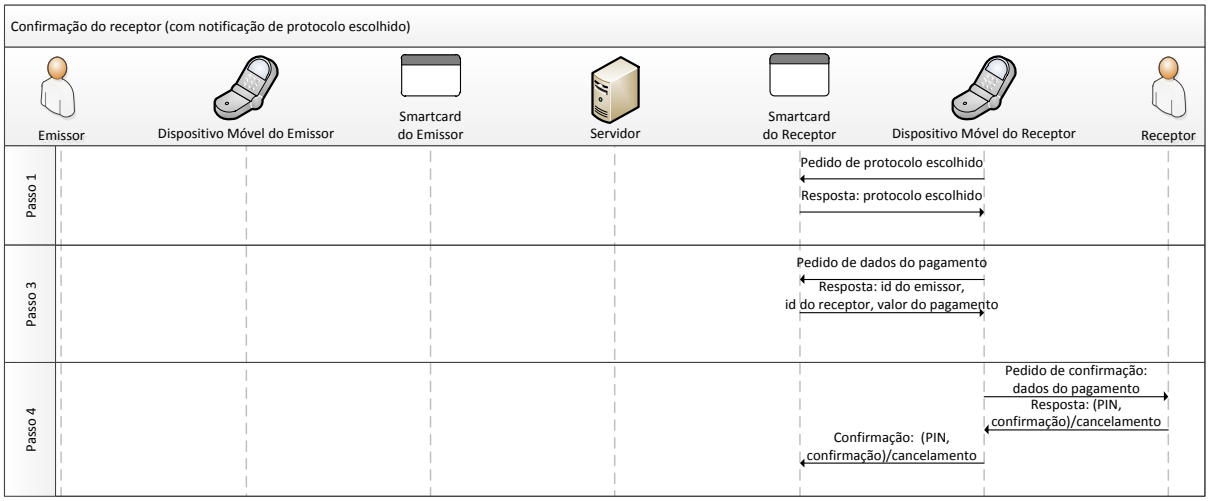


Figura 25. Passos definidos para a confirmação do receptor, no caso em que o dispositivo emissor notificou o dispositivo do receptor sobre o protocolo usado.

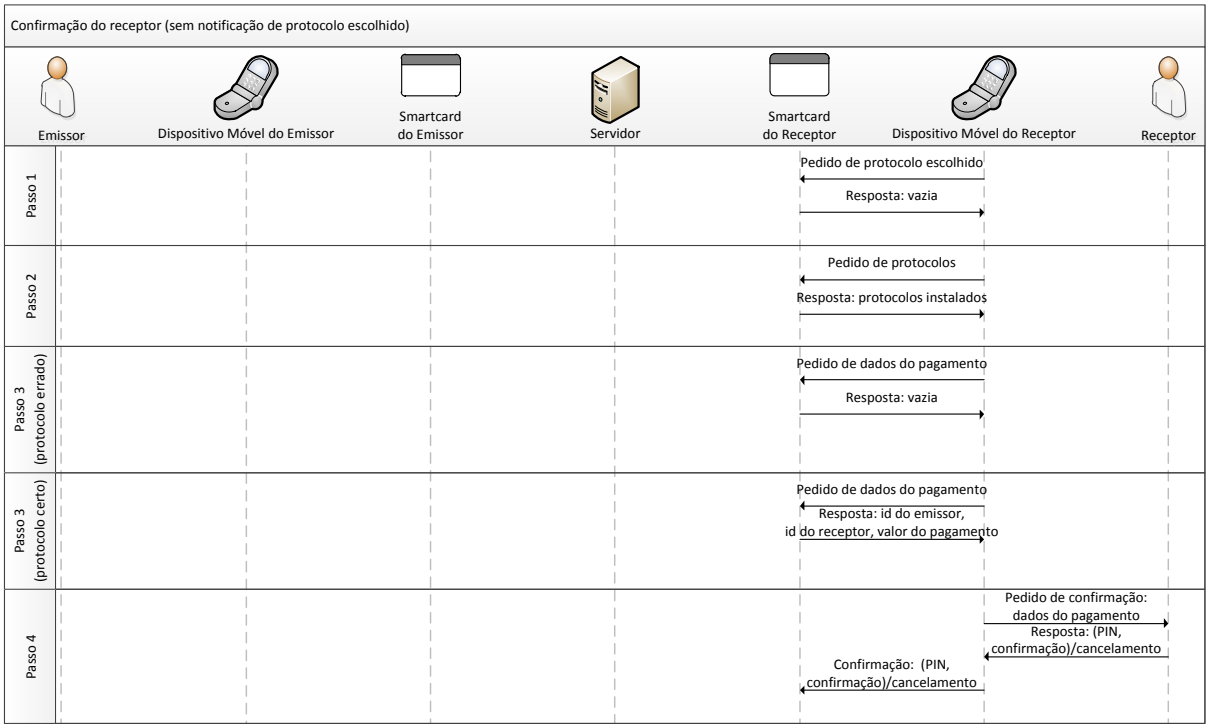


Figura 26. Passos definidos para a confirmação do receptor sem notificação do protocolo usado.

Fase de transferência: Se ambos os intervenientes aprovarem a transacção, é executada a última fase do pagamento denominada fase da transferência. Nesta fase é efectuado o conjunto de operações que representam o pagamento em si, e que tornam a transacção definitiva. No final desta fase, ambos os dispositivos indicam aos respectivos utilizadores o sucesso da transacção.

Fase de transferência (passo 1): No caso de um pagamento de proximidade, o emissor e o receptor devem aproximar de novo os dispositivos.

Fase de transferência (passo 2 a N - 1): O dispositivo do emissor executa os passos definidos pelo protocolo para a operação de transferência. Os componentes envolvidos, as mensagens trocadas, o número de passos N e a implementação de cada passo são dependentes do protocolo.

Fase de transferência (passo N): O dispositivo do emissor questiona o elemento seguro sobre o estado da transacção, para poder informar o emissor. No caso de um pagamento de proximidade, o dispositivo do receptor também executa o passo referido. As mensagens trocadas são independentes do protocolo. A implementação do passo é dependente do protocolo.

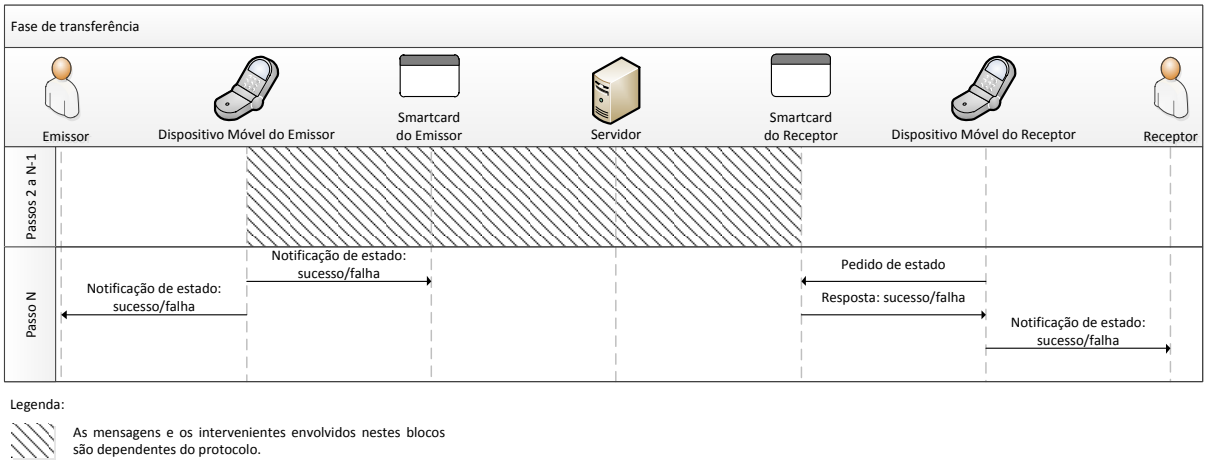


Figura 27. Passos definidos para o bloco fase de transferência.

3.3.3 Depósito

A operação de depósito permite aos utilizadores do sistema (clientes e vendedores) transformar créditos de um protocolo de pagamento em créditos de um sistema externo. Esta operação pode ser usada, por exemplo, para o dispositivo de um receptor informar uma instituição bancária de pagamentos que recebeu. É possível desenvolver protocolos de pagamento que não implementem a operação de depósito, se esta não for necessária ao funcionamento do protocolo. Este é geralmente o caso de protocolos *account-based*. No exemplo anterior, se o protocolo informasse a instituição bancária sempre que se processasse um pagamento, a operação de depósito podia ser eliminada.

O fluxo definido pelo sistema para uma operação de depósito é formado por duas acções (Figura 28). Na primeira (Pedido de Protocolos), o dispositivo do cliente determina quais os protocolos que conseguem executar depósitos. Na segunda (Execução de Depósito), o dispositivo processa os passos definidos para um depósito em cada um dos protocolos escolhidos no passo anterior.

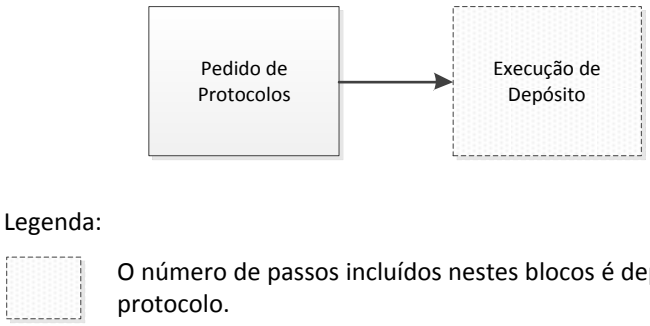


Figura 28. Fluxo da operação de depósito.

Os passos definidos pelo sistema para cada um dos blocos referidos são descritos de seguida.

Pedido de protocolos (passo 1): o dispositivo do cliente pede ao seu elemento seguro a lista de protocolos instalados. O módulo Registo de Protocolos é o responsável por responder ao pedido.

Execução de depósito (passos 1 a N): o dispositivo executa a operação de depósito de cada um dos protocolos da lista que suporte depósitos. Os componentes envolvidos, as mensagens trocadas, o número de passos N e a implementação de cada passo são dependentes dos protocolos.

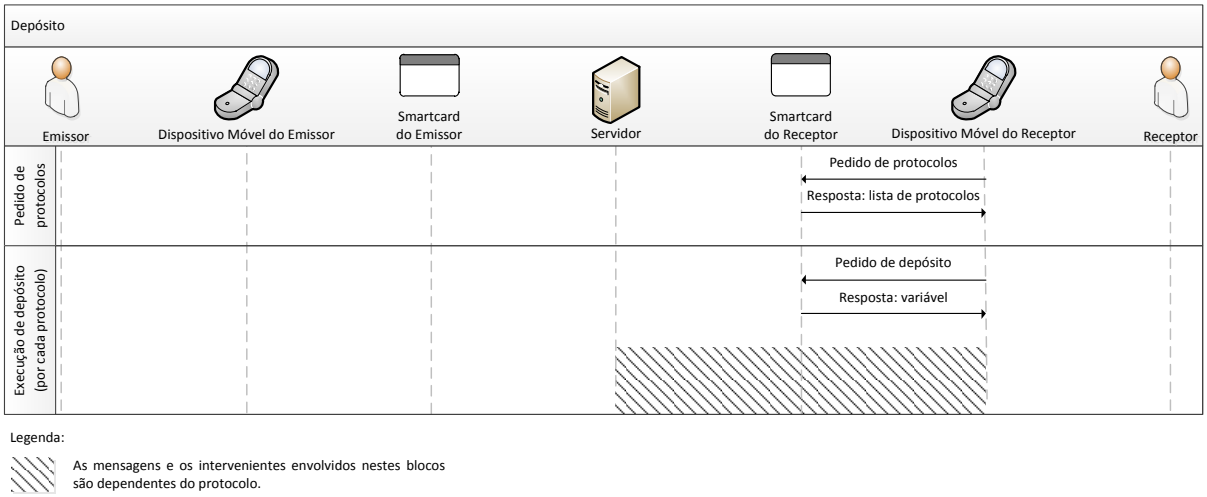


Figura 29. Passos definidos para a operação de depósito.

3.3.4 Verificação de saldo

A operação de verificação de saldo não é obrigatória para o funcionamento dos protocolos de pagamento. A sua função é mostrar ao utilizador da aplicação o saldo dos protocolos instalados.

A verificação de saldo consiste em duas acções representadas na Figura 30. Na primeira o dispositivo do cliente determina os protocolos que suportam esta operação. Na segunda acção o dispositivo apresenta ao cliente o saldo de cada protocolo que suporte a verificação de saldo.

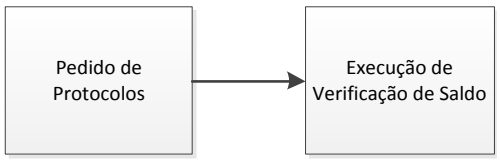


Figura 30. Fluxo da operação de verificação de saldo.

Os passos definidos pelo sistema para cada um dos blocos referidos são descritos de seguida (Figura 31).

Pedido de protocolos: o dispositivo do cliente pede ao seu elemento seguro a lista de protocolos instalados. O módulo Registo de Protocolos é o responsável por responder ao pedido.

Execução de verificação de saldo: o dispositivo do cliente pede o saldo actual a cada um dos protocolos da lista que suporte esta operação. As mensagens trocadas são independentes dos protocolos. A implementação do passo é dependente dos protocolos. O sistema ePaga define o saldo de um protocolo como o conjunto de dois valores: o saldo de pagamento e o saldo de depósito. O primeiro representa o valor disponível para efectuar pagamentos. O segundo representa o valor acumulado de pagamentos recebidos desde o último depósito. Este valor será utilizado no próximo depósito.

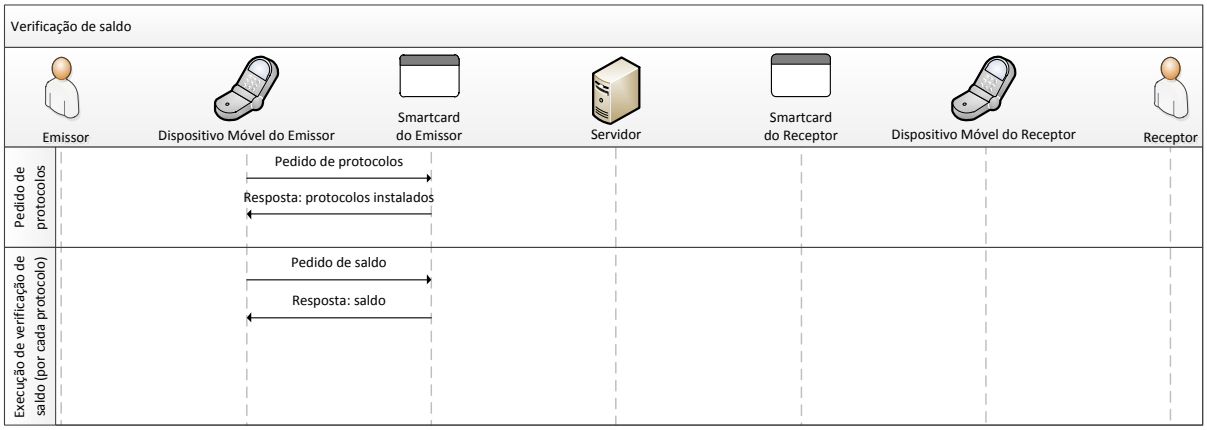


Figura 31. Passos definidos para a operação de verificação de saldo.

3.4 Protocolos

O sistema proposto tem de ser utilizado em conjunção com protocolos de pagamento desenvolvidos e instalados para o efeito. Para desenvolver um protocolo segundo o sistema ePaga, o programador tem de implementar o protocolo no elemento seguro e nos componentes remotos. A aplicação principal do cliente é totalmente independente do protocolo. Logo, o programador pode abstrair-se totalmente desta componente do sistema. Este componente pode inclusivamente ser implementado em dispositivos com linguagens e sistemas operativos diferentes, sem alterar o desenvolvimento dos protocolos.

O desenvolvimento de um protocolo compatível com o sistema ePaga é constituído por três passos descritos de seguida.

Implementação da aplicação cliente no elemento seguro: implementar as operações definidas pelo sistema que se adequam ao novo protocolo de pagamento.

Adição do novo protocolo à lista de protocolos instalados: atribuir valores aos atributos que caracterizam o protocolo, incluindo o fluxo de passos que representam cada uma das operações do protocolo. Este passo fornece a informação necessária para aplicação principal do cliente poder utilizar o novo protocolo, mais concretamente saber quais as operações em que o protocolo deve participar e saber como executar essas operações.

Implementação dos componentes remotos: implementar as operações que se adequam ao novo protocolo de pagamento, respeitando a interface definida pelo sistema.

Para desenvolver e testar a implementação do sistema que será descrita no capítulo 4, foi necessário implementar dois protocolos de pagamento. De seguida são descritas as arquitecturas destes protocolos.

3.4.1 Protocolo Token-based

O primeiro dos protocolos de pagamento implementados exemplifica um protocolo *token-based* e permite pagamentos de proximidade *offline*, nomeadamente transacções P2P e POS. O protocolo

representa uma versão simplificada dos conceitos usados no sistema fairCASH. Este sistema foi escolhido devido à simplicidade de implementação das técnicas e do fluxo do protocolo.

De seguida descreve-se a sequência de mensagens que representa o protocolo implementado. Nesta descrição o provedor de serviço remoto é denominado servidor.

Pedido de levantamento: o protocolo não usa a resposta ao pedido de levantamento para transportar dados.

Execução de Levantamento (passo 1): o dispositivo do emissor pede um *nonce* ao servidor (*nonce0*).

Execução de Levantamento (passo 2): o dispositivo do emissor pede ao elemento seguro interno que assine uma mensagem que represente a intenção de efectuar um levantamento. Este pedido inclui o *nonce0*. O elemento seguro responde com os campos recebidos, a cadeia de certificados que identifica o emissor[5], um *nonce* gerado por si (*nonce1)* e com uma assinatura. A assinatura abrange a quantia e ambos os *nonces*. Com a assinatura é garantida a integridade do campo quantia, além de ser evitado que esta mensagem, repetida posteriormente, seja aceite como outro pedido de levantamento.

$$Resposta = Quantia|Nonce0|Nonce1|Assinatura0_{Emissor}|Certificados_{Emissor}$$

$$Assinatura0_{Emissor} = \{H(Quantia|Nonce0|Nonce1)\}_{Priv_{Emissor}}$$

Execução de Levantamento (passo 3): o dispositivo do emissor usa a mensagem recebida do elemento seguro para fazer um pedido de levantamento ao servidor. Na resposta, o servidor envia o certificado que o identifica, os *tokens* que representam o levantamento e os dois nonces anteriores assinados por si. Com esta assinatura evita-se que esta mensagem, repetida posteriormente, seja aceite pelo emissor como outro levantamento valido. Os *tokens* e a assinatura são cifrados antes de serem enviados ao emissor.

$$Resposta = Certificado_{Servidor}|\{Tokens|Assinatura1_{Servidor}\}_{Pub_{Emissor}}$$

$$Assinatura1_{Servidor} = \{H(Tokens|Nonce0|Nonce1)\}_{Priv_{Servidor}}$$

Execução de Levantamento (passo 4): o dispositivo do emissor entrega a mensagem recebida ao seu elemento seguro.

Criação e pedido de serviço: a oferta de serviço contém apenas o valor do pagamento.

Negociação de pagamento (passo 1): se existir saldo suficiente para realizar o pagamento, o elemento seguro responde ao pedido de negociação com um *nonce* gerado por si (*nonce2)* e a cadeia de certificados que identifica o emissor.

$$Resposta = Quantia|Nonce2|Certificados_{Emissor}$$

Negociação de pagamento (passo 2): o dispositivo do emissor envia ao elemento seguro do receptor a mensagem recebida no passo anterior. O elemento seguro do receptor responde com a

[5] A cadeia de certificados que identifica um cliente é composta pelo seu certificado e pelo certificado da entidade que assinou o certificado do cliente.

cadeia de certificados que o identifica, um *nonce3* gerado por si e uma assinatura dos *nonces* 2 e 3. Esta assinatura garante ao emissor que está a negociar com quem o receptor afirma ser.

$$Resposta = Certificados_{Receptor}|Nonce3|Assinatura2_{Receptor}$$

$$Assinatura2_{Receptor} = \{H(Nonce2|Nonce3)\}_{Priv_{Receptor}}$$

Transferência (passo 1): o dispositivo do emissor pede ao seu elemento seguro os dados necessários para se processar a transferência. O elemento seguro responde com os *tokens* e uma assinatura, informação que representa a transacção. Esta assinatura representa a entrada no registo de transacções descrito no sistema fairCASH. Além da informação que caracteriza a transacção e os participantes, a assinatura engloba também o número de vezes que cada *token* foi utilizado ($Saltos\_Usados_{Tokens}$) e o *nonce3*. Os saltos são incluídos para que não sejam alterados por entidades não confiáveis. A inclusão do *nonce3* evita que esta mensagem, repetida posteriormente, seja aceite como outra transacção.

$$Resposta = Tokens|Assinatura2_{Emissor}$$

$$Assinatura2_{Emissor} = \{H(Quantia|Id_{Emissor}|Id_{Receptor}|Ids_{Tokens}|Saltos\_Usados_{Tokens}|Nonce3)\}_{Priv_{Emissor}}$$

Transferência (passo 2): o dispositivo do emissor entrega ao elemento seguro do receptor a informação que recebeu do seu elemento seguro do receptor.

Depósito (passo 1): o dispositivo do receptor pede um *nonce4* ao servidor.

Depósito (passo 2): o dispositivo do receptor faz um pedido de depósito ao seu elemento seguro. Este responde com a mensagem que representa o depósito. Esta mensagem inclui uma assinatura do *nonce4*. Esta assinatura garante que o receptor é o dono do certificado presente na mensagem. Os *tokens* e a assinatura são cifrados antes de serem enviados.

$$Resposta = Nonce5|\{Tokens|Assinatura3_{Receptor}\}_{Pub_{Servidor}}$$

$$Assinatura3_{Receptor} = \{H(Nonce4|Nonce5)\}_{Priv_{Receptor}}$$

Depósito (passo 3): o dispositivo do receptor envia ao servidor a informação que recebeu do seu elemento seguro.

3.4.2 Protocolo Account-Based

O segundo dos protocolos de pagamento implementados representa os protocolos de pagamento *account-based* para POS virtuais. Este protocolo baseia-se no protocolo proposto por Hassinen et al. para POS virtuais.

De seguida descreve-se a sequência de mensagens que representa o protocolo implementado.

Pedido de serviço: para além do valor do produto, o conteúdo da oferta de serviço inclui a descrição do produto.

Negociação de pagamento: o protocolo usa a negociação apenas para pedir o certificado ao vendedor.

Transferência: o protocolo segue as mensagens referidas nos passos 3 a 6 do protocolo para POS virtual proposto por Hassinen et al.

Transferência (passo 1): o dispositivo do cliente pede ao seu elemento seguro a mensagem que será utilizada no passo 2.

Transferência (passo 2): o dispositivo do cliente envia ao vendedor remoto a mensagem preparada pelo elemento seguro. Este passo equivale ao passo 3 "selecção de produto" do protocolo descrito na secção 2.6.8. Além do preço do produto ($Preço$), a mensagem inclui a identificação do produto ($Produto$), do cliente ($Id_{Cliente}$) e do banco a usar (Id_{Banco}). Inclui também um *timestamp* ($Data_{Cliente}$) e um *nonce* ($Nonce_{Cliente}$) gerados pelo dispositivo do cliente. O dispositivo do cliente adiciona ainda à mensagem uma assinatura dos campos anteriores que representa a autorização do cliente em relação à transferência.

$$\text{MSG} = \{\text{ORDER}|\text{SIG}_{\text{CUST}_{\text{BANK}}}\}\text{PKEY}_{\text{MERCH}} Mensagem = \{Encomenda|Assinatura_{Cliente_{Banco}}\}_{Pub_{Vendedor}}$$

$$\text{MSG} = \{\text{ORDER}|\text{SIG}_{\text{CUST}_{\text{BANK}}}\}\text{PKEY}_{\text{MERCH}}$$

$$Encomenda = Produto|Nonce_{Cliente}|Data_{Cliente}|\{H(Produto|Data_{Cliente})\}_{Priv_{Cliente}}|Id_{Cliente}|Id_{Banco}$$

$$Assinatura_{Cliente_{Banco}} = \{H(Data_{Cliente}|Id_{Vendedor}|Preço)|H(Produto|Nonce_{Cliente})\}_{SKEY_{CUST}}$$

Transferência (passo 3): Este passo equivale ao passo 4 "pedido de pagamento" do protocolo descrito na secção 2.6.8. O dispositivo do vendedor entrega ao dispositivo do banco um pedido de pagamento (Pedido) e autorizações do vendedor ($Assinatura_{Vendedor_{Banco}}$) e do cliente ($Assinatura_{Cliente_{Banco}}$). O pedido contém a informação necessária para executar o pagamento e para verificar as autorizações referidas. Esta informação é constituída pela identificação do vendedor ($Id_{Vendedor}$) e do cliente ($Id_{Cliente}$), pela selecção feita pelo cliente ($H(PRODUCT|NONCE_{CUST})$), pelo preço do produto e pelo *timestamp* gerado pelo dispositivo do cliente ($Data_{Cliente}$).

$$Mensagem = \{Pedido|Assinatura_{Vendedor_{Banco}}|Assinatura_{Cliente_{Banco}}\}_{Pub_{Banco}}$$

$$Pedido = Id_{Vendedor}|Id_{Cliente}|Data_{Cliente}|Preço|H(Produto|Nonce_{Cliente})$$

$$Assinatura_{Cliente_{Banco}} = \{H(Id_{Vendedor}|Id_{Cliente}|Data_{Cliente}|Preço|H(Produto|Nonce_{Cliente}))\}_{Priv_{Vendedor}}$$

Transferência (passo 3): Este passo equivale ao final do passo 5 "confirmação de pagamento" do protocolo descrito na secção 2.6.8. A mensagem enviada pelo dispositivo do banco para o dispositivo de vendedor contém informação que identifica o vendedor ($Id_{Vendedor}$)), o cliente ($Id_{Cliente}$)) e o produto ($H(PRODUCT|NONCE_{CUST})$). Contém também o valor do pagamento ($Preço$) e o *timestamp* gerado pelo dispositivo do cliente ($Data_{Cliente}$).

$$Mensagem = \{H(Id_{Vendedor}|Id_{Cliente}|Data_{Cliente}|Preço|H(Produto|Nonce_{Cliente}))\}_{Priv_{Banco}}$$

Transferência (passo 4): Este passo equivale ao passo 6 “entrega de produto” do protocolo descrito na secção 2.6.8. O dispositivo do vendedor encaminha a mensagem recebida para o dispositivo do cliente. Ambos os intervenientes podem verificar se os campos presentes na assinatura contêm os valores acordados, nomeadamente o valor do pagamento ($Preço$), o produto seleccionado ($Produto$) e os valores gerados pelo dispositivo do cliente ($Nonce_{Cliente}$ e $Data_{Cliente}$).

$$Mensagem = \{H(Id_{Vendedor}|Id_{Cliente}|Data_{Cliente}|Preço|H(Produto|Nonce_{Cliente}))\}_{Priv_{Banco}}$$

4 Implementação

Para demonstrar a arquitectura da solução proposta, foi implementada uma prova de conceito do sistema ePaga. Esta implementação é composta pela aplicação que contém o *middleware* proposto e dois protocolos de teste.

Para a implementação do sistema ePaga foi adoptada uma abordagem na qual os primeiros passos consistem em implementar e testar individualmente protótipos de protocolos que serão suportados pelo sistema. No passo seguinte estes protocolos servem de ponto de partida para a implementação de uma camada de apresentação unificada, seguida da camada de *middleware* (Figura 32).

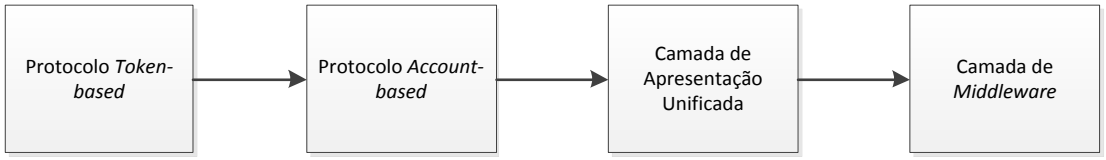


Figura 32. Fases de implementação da prova de conceito do sistema ePaga.

O desenvolvimento de cada uma das fases da implementação ilustradas na Figura 32 seguiu o seguinte fluxo (Figura 33). A *applet* residente no elemento seguro é o primeiro componente a ser implementado e testado. Após este componente estar terminado, é implementado o fragmento principal da aplicação cliente. Para garantir o correcto funcionamento do mesmo, este é inicialmente testado de forma isolada (Testes Unitários), antes de ser testado em conjunto com o componente anterior (Testes de Integração). O componente servidor é o último a ser implementado. À semelhança do componente que o precede, também este é testado primeiro isoladamente, e posteriormente em conjunto com a *applet* Java Card e com a *midlet* Java ME.

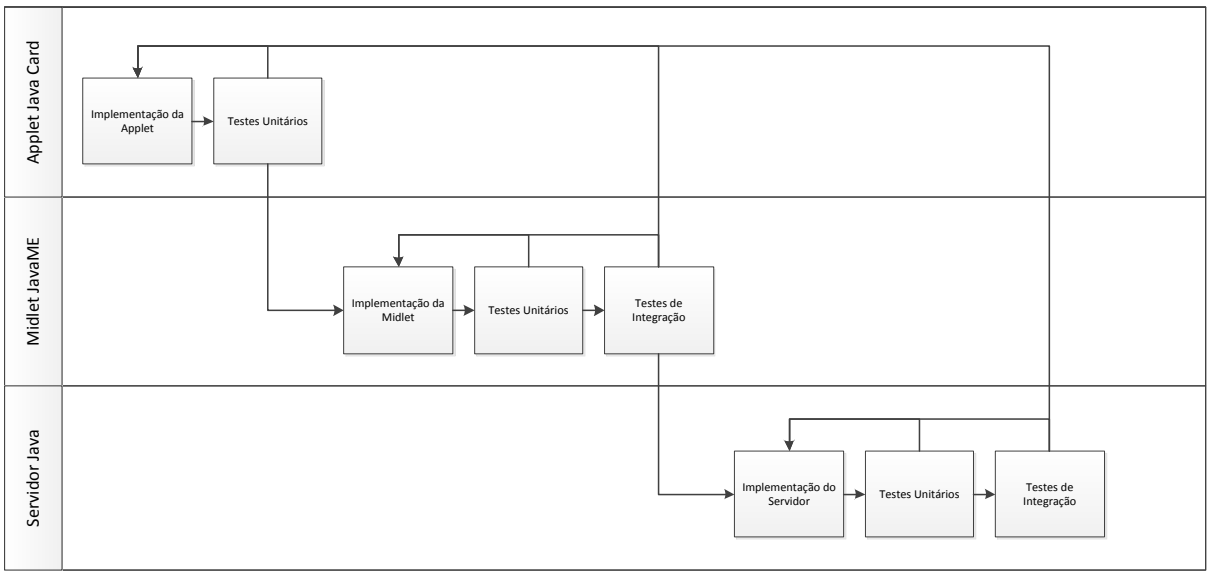


Figura 33. Fluxo da implementação de cada uma das funcionalidades desenvolvidas.

4.1 Componente JavaME

O componente Java ME (*Micro Edition*) da aplicação do cliente contém a camada de apresentação, os módulos de comunicação e parte do *middleware*. Este componente funciona como intermediário entre o utilizador, o módulo Java Card e componente servidor.

A leitura de informação de pagamento de forma automática, a partir da interacção com tags NFC ou páginas web de pagamento, não é essencial para demonstrar funcionamento do sistema ePaga. Como tal, apesar de ser prevista pela arquitectura, não foi implementada nesta prova de conceito.

4.1.1 Camada de Apresentação

Cabe à camada de apresentação o pelouro de interagir com o utilizador. Os pedidos do utilizador são encaminhados para a camada de execução. O principal desafio nesta camada é manter uma interface constante ao longo da utilização de protocolos distintos. Para atingir este objectivo, o código desta camada deve ser o mais genérico possível. Nos casos em que não é possível uma abstracção completa do protocolo em uso, a camada de apresentação utiliza os metadados fornecidos pela camada de execução, para interagir com o utilizador de um modo que faça sentido para o protocolo utilizado. No resto desta secção são apresentados os pontos de interacção da aplicação com o utilizador.

Na Figura 34 é apresentado o ponto de partida para todas as operações da aplicação.

Para efectuar um levantamento (Figura 35), o utilizador deve seguir os seguintes passos:

- Escolher o protocolo que deseja “carregar”.
- Introduzir a quantia que deseja levantar e confirmar o levantamento.

Figura 34. Ecrã inicial.

Figura 35. Ecrãs mostrados durante uma operação de levantamento.

Os passos para efectuar um pagamento de proximidade são os seguintes (Figura 36):

- O receptor introduz a quantia do pagamento.
- O emissor e o receptor aproximam os dispositivos.
- O emissor e o receptor verificam e confirmam os dados da transacção
- O emissor e o receptor autorizam o pagamento através da introdução dos respectivos PINs.
- O emissor e o receptor aproximam novamente os dispositivos.

Figura 36. Ecrãs mostrados durante uma operação de pagamento de proximidade.

Para efectuar um pagamento à distância (Figura 37), o utilizador deve seguir os seguintes passos:

- Introduzir o URL do serviço.
- Verificar e confirmar os dados da transacção.
- Autorizar o pagamento através da introdução do PINs.

Figura 37. Ecrãs mostrados durante uma operação de pagamento à distância.

Para executar uma operação de verificação de saldo (Figura 38) ou de depósito (Figura 39), a única acção necessária resume-se a escolher a respectiva opção no ecrã inicial.

No caso de um passo de uma operação terminar com erro, a aplicação apresenta ao utilizador um ecrã como o representado na Figura 40. Este ecrã oferece ao utilizador a possibilidade de requisitar que o passo seja repetido, ou que a operação seja cancelada. Cabe a cada protocolo suportar a repetição de passos e o cancelamento de operações.

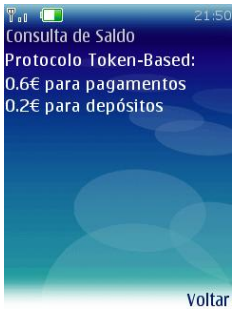


Figura 38. Ecrã de consulta de saldo.

Figura 39. Ecrã de depósito.

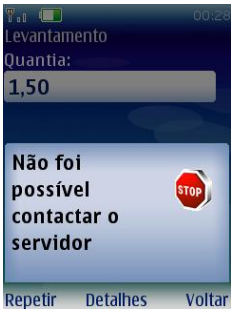


Figura 40. Exemplo de um ecrã de erro.

4.1.2 Camada de Execução

A camada de execução representa o cérebro da *midlet*. Tem como função recolher dados das camadas de *web services* e de *smartcard* para processar comandos da camada de apresentação. A Figura 41 ilustra a estrutura da camada explicada de seguida.

A classe denominada PaymentManager é simultaneamente o ponto de entrada e o cerne da camada. As restantes classes dividem-se em dois grupos com objectivos distintos. As do primeiro grupo fornecem informação às decisões do PaymentManager. As do segundo fornecem informação ao elemento seguro interno. O primeiro grupo é composto pelas classes Protocol, Operation e OperationStep.

A classe Protocol é formada pelos metadados que representam um protocolo de pagamento. Os metadados são os seguintes:

- Um identificador do protocolo perante o sistema.
- Um identificador da *applet* do protocolo.
- O nome para identificar o protocolo perante o utilizador.
- As operações suportadas pelo protocolo.
- A disponibilidade do protocolo em participar em pagamentos de proximidade, à distância e *offline*.

A classe Operation define os passos que compõem uma operação de um protocolo de pagamento.

A classe OperationStep representa um passo de uma operação. Esta classe contém informação que especifica como um passo deve ser executado, através dos seguintes atributos:

- O tipo de componente de destino da mensagem que concretiza o passo. Em linha com o que foi descrito no capítulo de arquitectura, este tipo pode ser um elemento seguro interno ou externo, um vendedor remoto ou um servidor.
- O endereço do destino da mensagem. Este atributo é utilizado apenas nos tipos vendedor remoto e servidor.
- Um código que permite ao destinatário identificar o passo em execução.

O segundo grupo contém as classes Time e Context. A classe Time define um formato para o objecto que introduz o conceito de tempo no elemento seguro. A classe Context representa a informação de contexto que auxilia o elemento seguro a escolher o protocolo de pagamento a usar.

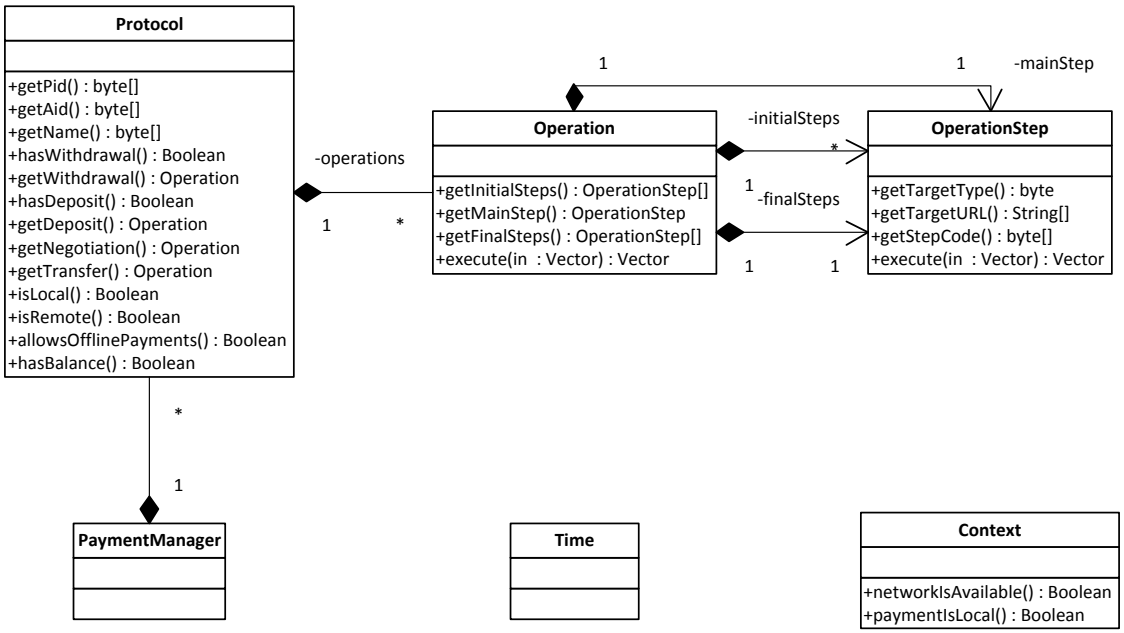


Figura 41. Diagrama de classes simplificado da camada de execução.

4.1.3 Camadas de Smartcard e Web Services

As camadas *smartcard* e *web services* concretizam as decisões tomadas pela camada de execução.

A camada smartcard age como uma abstracção do acesso a *smartcards contactless*. A comunicação com o módulo Java Card, quer seja do mesmo dispositivo, ou de outro dispositivo NFC, usa uma ligação ISO14443.

A comunicação remota da aplicação foi implementada com *web services*. Os *stubs* necessários para aceder aos *web services* foram gerados pela ferramenta *Stub Generator* incluído no *Sun Java Wireless Toolkit*. Foi usada a versão 2.5.1 do *Toolkit*.

4.2 Componente Java Card

O componente Java Card contém a parte crítica da aplicação cliente, nomeadamente a lógica de escolha de protocolos, o armazenamento de metadados e os protocolos de pagamento. O sistema confia na execução segura e armazenamento protegido contra intrusões deste componente. No resto desta secção descrevem-se os vários módulos deste componente: *middleware* e protocolos.

4.2.1 Middleware

Este módulo compõe a estrutura que permite à aplicação usufruir dos protocolos de pagamento. O módulo é formado pelas classes descritas de seguida.

A classe AbstractApplet disponibiliza funções genéricas comuns a várias *applets* da aplicação, como o tratamento básico de mensagens.

A *applet* PaymentInfoApplet implementa as funcionalidades de criação de serviço e pedido de serviço. A PaymentInfoApplet guarda a informação de serviço criada pelo receptor do pagamento, até que o emissor contacte esta *applet* para recolher esta informação.

A *applet* ProtocolChooserApplet expõe a funcionalidade de escolher os melhores protocolos para um determinado pagamento. A ProtocolChooserApplet implementa a filtragem de protocolos disponíveis para um pagamento. Por outro lado, a funcionalidade de ordenamento de protocolos é delegada à classe PolicyManager. É também delegado o carregamento dos protocolos instalados à classe ProtocolsDataSIO.

A classe PolicyManager oferece uma interface de acesso à política de ordenação de protocolos, para ser acedida pela ProtocolChooserApplet. O objectivo da existência desta classe é separar os mecanismos de escolha de protocolos das políticas de ordenação de protocolos. Dada uma lista de protocolos, o PolicyManager entrega cada um dos protocolos à política em uso. No final o PolicyManager pede à política o protocolo mais indicado da lista.

A classe AbstractPolicy serve de base para todas as políticas do sistema. Para demostrar o conceito do sistema ePaga foi implementada a política LowestCostPolicy. Este exemplo de política escolhe o protocolo que apresente um custo menor para executar o pagamento. A razão da interface escolhida para as políticas prende-se com o objectivo de as dotar com o máximo de liberdade possível. A LowestCostPolicy não precisa de ocupar memória com listas de protocolos, ou de ter a lista completa de protocolos antes de calcular o melhor candidato. No entanto, esta interface acomoda futuras políticas que exijam os referidos requisitos.

A *applet* ProtocolsDataApplet fornece à aplicação um modo de aceder à lista de protocolos instalados. A implementação desta funcionalidade é delegada à classe ProtocolsDataSIO. A ProtocolsDataApplet disponibiliza também a funcionalidade de manter a informação de qual protocolo está a ser utilizado num dado momento. O dispositivo do emissor acede à ProtocolsDataApplet do receptor para a informar do protocolo que escolheu. O fragmento principal da aplicação do receptor recupera posteriormente esta informação, como foi descrito no capítulo da arquitectura.

A classe ProtocolsDataSIO tem a responsabilidade de manter a lista de protocolos instalados. A funcionalidade referida não foi incluída na ProtocolsDataApplet porque a lista de protocolos é necessária para o funcionamento de duas *applets*, a ProtocolsDataApplet e a ProtocolChooserApplet. Esta classe foi implementada como um *shareable interface object*, uma forma de partilhar informação entre *applets*.

A classe Protocol contém os metadados que representam um dos protocolos instalados. Os dados que esta classe e a classe Protocol da *midlet* contêm são exactamente os mesmos. A diferença reside nos dados que são utilizados em cada um dos casos. No fragmento Java Card da aplicação

não são utilizados atributos como o nome do protocolo. Por outro lado, são utilizados dois atributos que foram omitidos na descrição do fragmento Java ME. Esta classe contém um atributo que permite calcular o custo de um pagamento, assim como o valor máximo suportado pelo protocolo para um pagamento.

A classe Context representa a informação de contexto que ajuda o *middleware* a escolher os protocolos que devem ser usados. Esta classe agrega a informação sobre o pagamento gerada pelo dispositivo do emissor e pelo dispositivo do receptor. No conjunto de informação proveniente do receptor encontram-se dados extra, que não são interpretados pelo *middleware*, com excepção da política utilizada. Estes dados foram adicionados para acomodar variáveis disponibilizadas pelo vendedor e que influenciem a decisão tomada pela política.

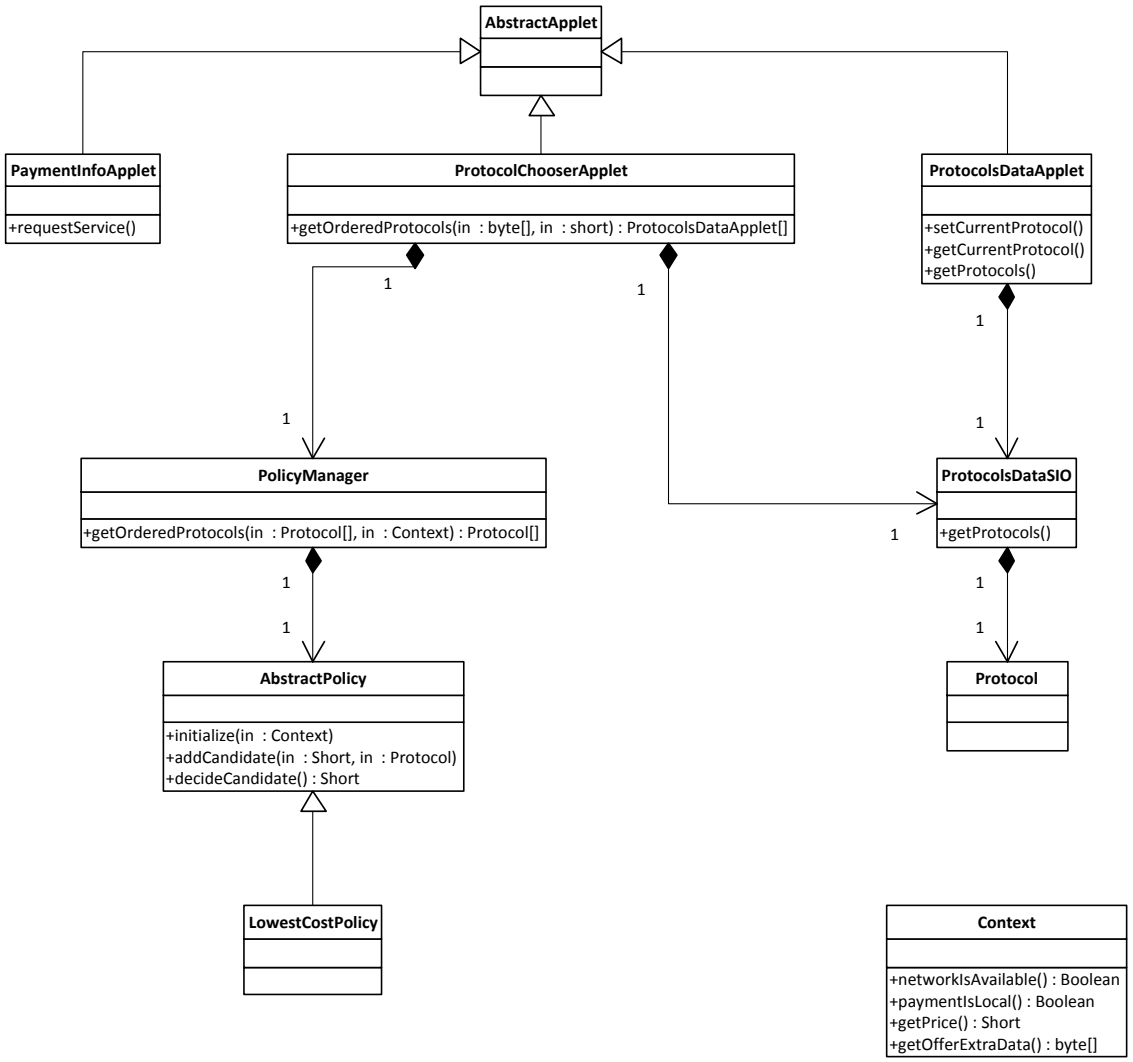


Figura 42. Diagrama de classes simplificado do módulo de *middleware*.

4.2.2 Protocolos

Nesta camada são implementados os protocolos de pagamento utilizados pelo sistema. Cada protocolo de pagamento inclui uma *applet* que implementa a lógica do protocolo. De modo a forçar a sequência normal de comandos que representa o protocolo de pagamento, a *applet* mantém o estado

da operação que está a executar. São recusados comandos que não pertençam à sequência normal do protocolo. A aplicação desenvolvida disponibiliza classes base que simplificam a implementação de protocolos, através da eliminação de código repetido.

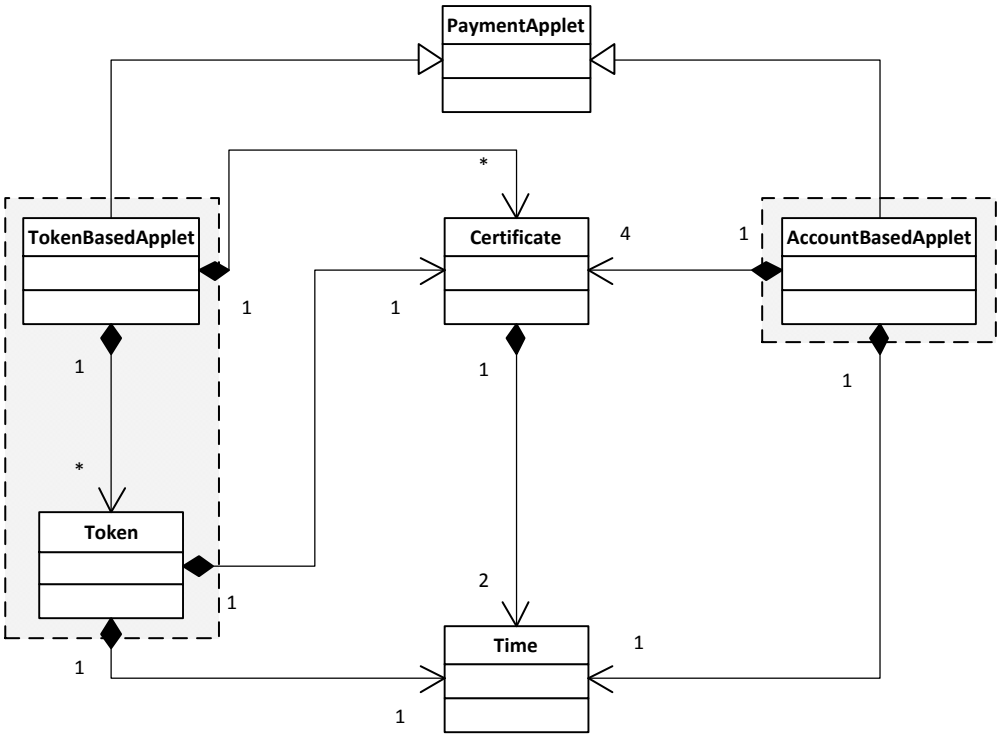
A classe PaymentApplet implementa funções comuns aos vários protocolos, como verificação de PIN ou geração de nonces.

A representação de um certificado, implementada pela classe Certificate, é formada pelos seguintes atributos: um tipo, um identificador, uma chave pública, uma data de validade, uma assinatura e um identificador do assinante. A assinatura abrange o tipo, o identificador, a chave pública e a validade. O atributo tipo determina se a entidade é anónima e que operações lhe são permitidas.

Na implementação do protocolo *token-based* foi adoptada uma cadeia de certificação simplificada em relação ao sistema fairCASH. Esta cadeia contém os seguintes componentes: entidade de raiz, servidor e cliente.

A entidade de raiz é a única em que todos os participantes confiam. A chave pública desta entidade tem de estar presente quer nos servidores, quer nos clientes. Este certificado é usado para verificar a assinatura dos certificados dos servidores. Os certificados dos servidores são usados para verificar a assinatura dos *tokens* e certificados de clientes. São também usados para verificar a autenticidade de mensagens enviadas pelos servidores. Os certificados dos clientes são usados para verificar a autenticidade de mensagens enviadas pelos clientes.

O protocolo *token-based* exigiu a implementação de uma classe específica para além da *applet*, a classe que representa um *token*. Esta é constituída pelos seguintes atributos: um identificador, um valor monetário, um número de saltos permitidos, uma data de validade, uma assinatura, um número de saltos usados e um identificador do assinante. A assinatura abrange o identificador, o valor, os saltos permitidos e a validade.



Legenda:

As classes destes blocos representam implementações de protocolos de pagamento.

Figura 43. Diagrama de classes simplificado do módulo dos protocolos.

Para as operações criptográficas executadas no elemento seguro do cliente e no servidor, foi usado o algoritmo RSA (Rivest, Shamir e Adleman) de acordo com a versão 1.5 dos *standards* PKCS #1 (*Public-Key Cryptography Standards*). O tamanho adoptado para as chaves do algoritmo foi de 512 bits. Nas assinaturas digitais foi ainda usado o algoritmo de *hashing* SHA-1 (*Secure Hash Algorithm*).

4.3 Componentes Remotos

Como foi mencionado anteriormente, o sistema ePaga define interfaces para o vendedor remoto e o servidor (Figura 44). Visto serem interfaces genéricos comuns a todos os protocolos de pagamentos, estes têm de ser o mais abrangentes possível. Assim, todos os métodos definem como dados de entrada e saída uma cadeia de bytes codificada em base 64. Deste modo prevêem-se quaisquer tipos de dados utilizados pelos protocolos. Para além dos métodos básicos esperados num protocolo de pagamento, os interfaces de ambos os componentes definem um método extra (auxMethod), cujo comportamento é indefinido. Assim, se os métodos definidos não forem suficientes para prever o comportamento do protocolo, o protocolo implementa este método.

Uma implementação do vendedor remoto deve especificar o tratamento das mensagens negociação de pagamento (exchangeNegotiationInfo), transferência (exchangeTransferInfo) e pedido de serviço (requestService), ao qual deve responder com, no mínimo, o valor do pagamento. Uma

implementação do servidor deve especificar o tratamento das mensagens de levantamento (withdraw) e depósito (deposit).

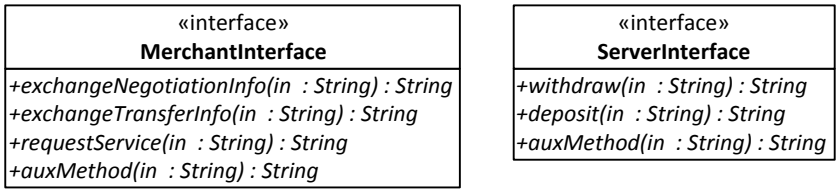


Figura 44. Interfaces definidas para os componentes vendedor remoto (MerchantInterface) e servidor (ServerInterface).

Estas interfaces apenas especificam a interacção dos componentes vendedor remoto e servidor com a aplicação cliente. Outras interacções podem desviar-se destas interfaces. No protocolo *account-based*, a interacção entre o vendedor remoto e o servidor é um exemplo desta situação. Tal como foi descrito no capítulo da arquitectura, o vendedor comunica com o servidor durante o pagamento. Esta operação foi implementada através de um *web service* não especificado pelo sistema ePaga.

Para representar a entidade central do protocolo *token-based* foi implementado um servidor que suporta as funcionalidades descritas na arquitectura, nomeadamente emitir novos *tokens* e receber depósitos de *tokens*. Para além destas funções, o servidor representa ainda a entidade de raiz do sistema e emite certificados de clientes. Este componente usa a chave privada correspondente para assinar o certificado de servidor, com que executa as restantes operações.

No sistema fairCASH, os levantamentos e depósitos estão associados a uma transferência de fundos para ou do sistema respectivamente. Estes passos foram ignorados na implementação do protocolo *token-based*.

Para o protocolo account-based foi implementado um vendedor remoto e um servidor com as funções descritas na arquitectura do protocolo. Tal como no protocolo *token-based*, o servidor emite certificados de clientes e vendedores.

Os componentes vendedor remoto e servidor foram implementados na versão 1.6 da linguagem Java. Para as funções criptográficas foi utilizada a versão 1.46 da biblioteca Bouncy Castle.

4.4 Ambiente de Desenvolvimento

Para o desenvolvimento do módulo Java ME do componente cliente foi escolhido o SDK (*Software Development Kit*) do Nokia 6212. Este SDK foi escolhido por ser a solução de desenvolvimento para NFC disponível mais completa. O SDK inclui um simulador que permite testar o módulo Java ME.

O módulo Java Card foi implementado com a utilização da versão 2.2.2 do *Java Card Development Kit*. Para a simulação deste módulo foram usados os simuladores incluídos no *Development Kit*: CREF (*C-language Java Card RE*) e JCDWE *(Java Card platform Workstation Development*).

Para testar os módulos Java ME e Java Card em conjunto foi utilizado um *plugin* do SDK que, da perspectiva do simulador do telemóvel, representa um *smartcard*. Este *smartcard* pode ser acoplado

à antena NFC do telemóvel simulado, ou utilizado como elemento seguro do telemóvel. O *plugin* recebe comandos do simulador do telemóvel e envia-os ao simulador de Java Card.

5 Avaliação

Esta tese tem como objectivo a criação de um protótipo funcional que implemente a arquitectura proposta. Esta implementação inclui o desenvolvimento do *middleware* referido e da camada de apresentação. Para verificar o cumprimento dos objectivos traçados para o sistema ePaga, a implementação desenvolvida foi avaliada segundo critérios qualitativos, cujo cumprimento não pode ser quantificado, e critérios quantitativos, cujo cumprimento pode ser objectivamente demonstrado através de testes à aplicação.

5.1 Avaliação Qualitativa

De seguida analisa-se o desempenho do sistema em relação a cada requisito referido na secção 1.2.

Consistência: o desenvolvimento de uma interface consistente implica a existência de transparência em relação ao facto de estarem a coexistir vários protocolos no mesmo dispositivo. Essa transparência foi atingida no pagamento e no depósito, visto que nestas operações o utilizador da aplicação, que seja o emissor ou receptor de pagamentos, não tem de tomar nenhuma decisão em relação ao protocolo que deve ser utilizado. O utilizador pode concluir estas operações sem ter a noção de qual protocolo foi utilizado. Este objectivo não foi completamente alcançado nas operações de levantamento e na verificação de saldo. No levantamento o cliente tem de escolher o protocolo que deseja "carregar", enquanto na verificação de saldo o valor do saldo é descriminado por protocolo. Nestas operações optou-se por dar mais informação e controlo ao utilizador sobre a aplicação em detrimento da consistência.

Interoperabilidade: a concentração dos protocolos e do estado da aplicação no elemento seguro leva a uma maior independência em relação ao dispositivo, nomeadamente ao nível do sistema operativo do dispositivo e respectiva linguagem de programação. O sistema também não limita o controlo dos componentes remotos a um determinado actor como um banco ou uma operadora. Esta distribuição de responsabilidades por actores é deixada em aberto. A possibilidade de interacção entre um dispositivo que usa o sistema ePaga e outro sem o sistema instalado também não foi limitada pela arquitectura. Se um emissor que use outra aplicação e um receptor com a aplicação ePaga desejarem executar um pagamento, a interacção difere da normal em dois pontos. O primeiro centra-se no passo de pedido de serviço. Em vez de entregar este pedido ao respectivo módulo do *middleware*, o dispositivo do emissor comunica directamente com o protocolo. Para resolver situações como esta, no passo de oferta de serviço, o dispositivo do receptor informa todos os protocolos sobre a oferta de serviço. Deste modo qualquer um deles pode responder ao pedido de serviço. O custo desta funcionalidade reflecte-se numa oferta de serviço mais lenta, como será demonstrado na secção 5.2. O segundo reside no facto de o dispositivo do emissor não informar explicitamente o dispositivo do receptor sobre o protocolo que esta a usar. No entanto, nesta situação a aplicação do receptor percorre todos os protocolos instalados, de modo a encontrar o protocolo usado. Assim a operação é bem executada com sucesso, com o custo de uma transacção mais lenta.

A situação inversa também é possível. Se o emissor do pagamento utilizar a aplicação ePaga e o receptor do pagamento usar outra aplicação, a principal diferença centra-se no passo de pedido de serviço. Como o dispositivo do emissor não consegue contactar o *middleware* para executar um pedido de serviço, deverá contactar directamente os protocolos que suporta, até descobrir se um deles está pronto para ser usado. Esta funcionalidade não foi implementada na prova de conceito, pelo que este protótipo não atinge todo o potencial que a arquitectura possibilita no capítulo da interoperabilidade.

Universalidade: a arquitectura cumpre este requisito porque as operações definidas permitem que quaisquer dos tipos de pagamentos abordados sejam implementados sobre o sistema. No entanto, a prova de conceito implementada não atinge a mesma universalidade. A razão para esta disparidade prende-se com o tempo necessário para implementar protocolos para todas as situações previstas. Ainda assim, os protocolos implementados demonstram que o sistema consegue executar tipos completamente distintos de pagamento, como P2P e POS virtual.

Simplicidade: apesar de este requisito ter condicionado o desenho do sistema e a implementação da prova de conceito, não foi possível medir o seu cumprimento de forma objectiva. O ambiente de simulação em que foi desenvolvido o sistema invalida testes com utilizadores reais. Na oferta de serviço descrita na secção 3.3, a prova de conceito implementada não atinge a simplicidade prevista na arquitectura, visto que apenas permite a introdução manual dos dados do serviço.

5.2 Avaliação Quantitativa

Para analisar de forma quantitativa o desempenho da prova de conceito implementada, foram realizadas medições sobre três aspectos: tempo de processamento das operações, dados transmitidos e dimensão das aplicações. As operações executadas nos testes consistiram num levantamento de 50 cêntimos, um pagamento local de 20 cêntimos, um pagamento remoto de 20 cêntimos e um depósito de 20 cêntimos.

A inclusão de um protocolo numa aplicação genérica em substituição uma aplicação específica introduz uma diminuição da eficiência, que se reflecte tanto no tempo de processamento, como na quantidade de dados transmitidos. Um dos requisitos no desenvolvimento da aplicação ePaga é minimizar este *overhead*.

Para analisar o *overhead* introduzido na rapidez de execução das operações dos protocolos, foi medido o tempo que a aplicação gasta a processar cada operação (Figura 45). Visto que os testes foram executados num ambiente simulado, os tempos de comunicação não têm qualquer relação com valores medidos num ambiente real. Assim, os tempos medidos aproximam-se de tempos de processamento. Pela mesma razão, os valores absolutos analisados isoladamente não têm um significado relevante. No entanto, a relação entre os valores obtidos pela aplicação ePaga e a aplicação dedicada pode ser transportada para um ambiente real. Na Figura 45 observa-se que a recepção de pagamento se destaca pela negativa, em relação às restantes operações, com um *overhead* próximo dos 50%. Este resultado deve-se a dois factores. A aplicação ePaga define dois passos para esta operação que a aplicação dedicada não contemplava, nomeadamente o pedido de

serviço explícito e a detecção do protocolo que está a ser usado. Estes passos adicionais são os principais responsáveis pela disparidade nos valores observados para esta operação. O segundo factor que leva ao destaque deste resultado está relacionado com o tempo de execução reduzido da operação. A recepção de pagamento é a operação com um tempo de processamento mais curto, o que faz com que o mesmo *overhead* introduzido pelo *middleware* se note mais nesta operação do que em qualquer outra.

Com a finalidade de adicionar algum contexto a estes valores, é interessante combiná-los com um exemplo de um protocolo de pagamento existente que tenha sido testado num ambiente mais próximo do real. Em [7] os autores apresentam uma média de 7,5 segundos para executar os passos de um pagamento local, incluindo tempos de processamento e comunicação. Através da soma dos valores das operações pagamento local e recepção de pagamento da Figura 45, obtém-se o tempo de processamento de um pagamento local. A partir deste cálculo para a aplicação ePaga e a aplicação dedicada, obtém-se um valor de 30% para o *overhead* do *middleware* num pagamento local. Se fosse implementado na prova de conceito da aplicação ePaga o protocolo avaliado em [7], estima-se que o valor médio do tempo de processamento seria inferior a 9,8 segundos. Este valor corresponde ao pior caso possível, em que se assume um tempo de comunicação nulo e, consequentemente, um tempo de execução igual ao tempo de processamento. Este é o pior caso porque os 30% de *overhead* são aplicados ao tempo de processamento, que nesta situação atingiria o seu valor máximo.

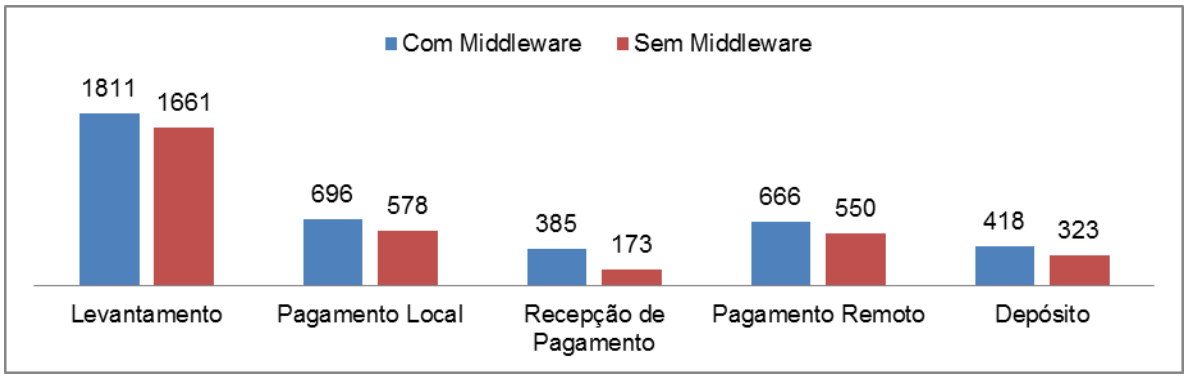


Figura 45. Representação gráfica do tempo de processamento (em milissegundos) das principais operações definidas pelo sistema. Em cada par de colunas, a coluna impar indica o tempo relativo à aplicação ePaga e a coluna par indica o tempo conseguido pela aplicação dedicada do protocolo. Para a operação de pagamento remoto foi utilizado o protocolo *account-based*. Para as restantes operações foi utilizado o protocolo *token-based*. Cada valor apresentado resulta de uma média de cinco amostras.

Foi analisado o *overhead* introduzido na quantidade de dados transmitidos entre o dispositivo móvel e os elementos seguros interno e externo. Entre comunicação local e remota, este é o tipo de comunicação em que o *middleware* conduz à transmissão de mais dados. No entanto, este é também o tipo que tem menos influência no desempenho do protocolo, quer no custo da transacção, quer no tempo de execução da transacção. Os resultados desta análise estão ilustrados na Figura 46.

Assim como no tempo de processamento, a prova de conceito implementada volta a mostrar um valor excessivo na operação de recepção de pagamento.

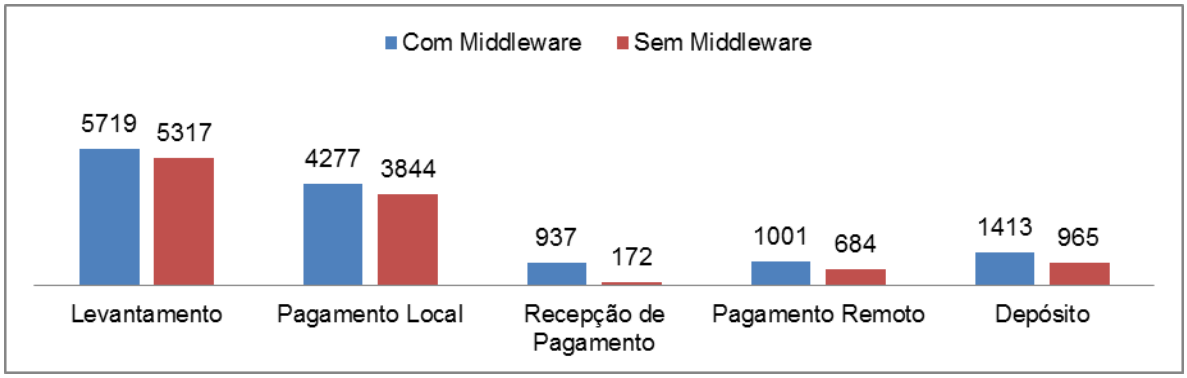


Figura 46. Representação gráfica da quantidade de dados (em *bytes*) transmitidos, para cada operação, entre o dispositivo móvel e os elementos seguros interno e externo.

Nos dispositivos móveis, a quantidade de dados transmitidos através da rede da operadora tem influência directa no custo da operação, pelo que esta grandeza também foi medida. Pode observar-se na Figura 47 que a diferença introduzida pela aplicação ePaga não é significativa. Esta semelhança entre as aplicações justifica-se com o número reduzido de dados e passos adicionais introduzido pelo *middleware* neste tipo de comunicação. De facto, a principal origem do *overhead* na comunicação remota é a codificação mais ineficiente a que uma aplicação genérica conduz. As aplicações dedicadas têm um maior conhecimento sobre os tipos de informação que o protocolo produz, enquanto aplicação ePaga trata os dados uniformemente.

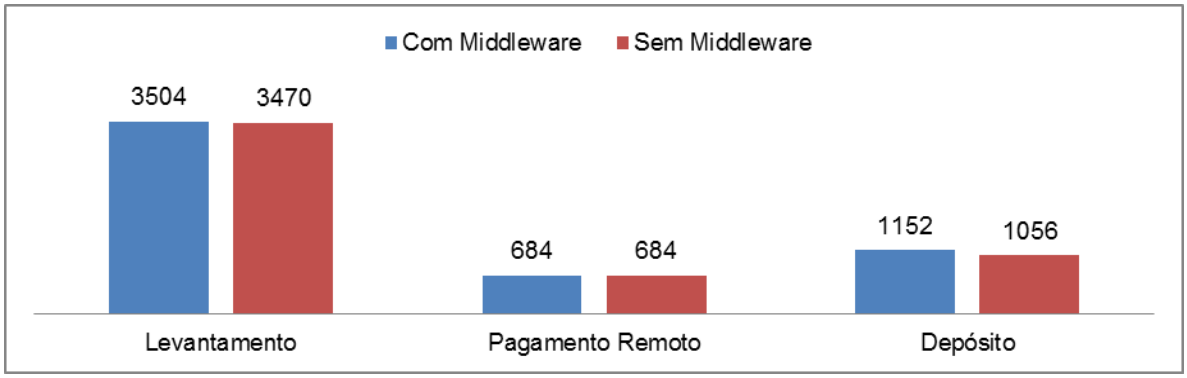


Figura 47. Representação gráfica da quantidade de dados (em *bytes*) transmitidos, para cada operação, entre o dispositivo móvel e os componentes remotos. Não foram incluídas no gráfico as operações que não geraram este tipo de trafego. Apenas foram contabilizados os dados transmitidos pela aplicação. Foram excluídos os dados formam os cabeçalhos da pilha de protocolos que suporta os *web services* utilizados.

De modo a analisar o esforço de implementação necessário para desenvolver as aplicações, foi medido o número de linhas de código de cada uma. Estes valores estão ilustrados na Figura 48, na qual se observa que cerca de metade das linhas escritas são escritas em JavaME. Se por um lado este código é em geral mais fácil de escrever, por outro lado é o código menos relevante para a lógica do protocolo. Deste modo, a utilização da aplicação ePaga e do respectivo *middleware* reduz o esforço de implementação de um protocolo. É também relevante mencionar que o código disponibilizado pelo *middleware,* com o intuito de servir de base para os protocolos, representa outra contribuição para simplificar o código.

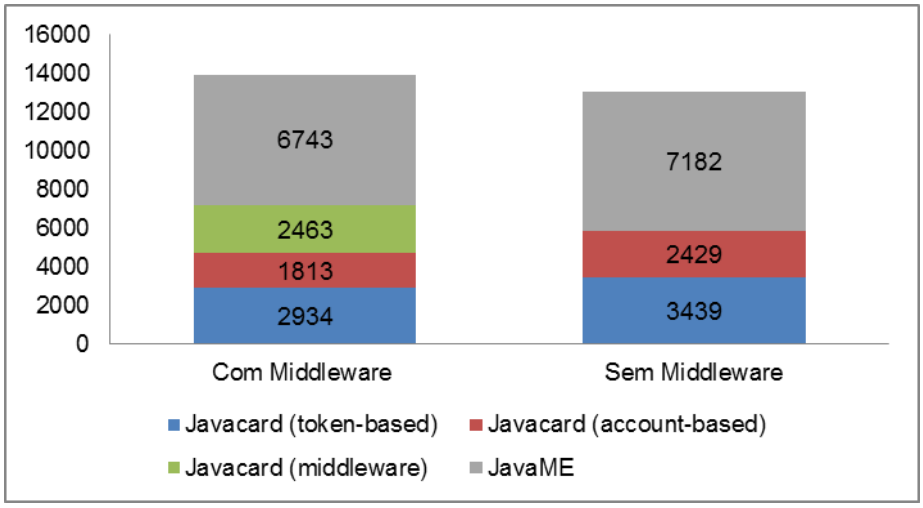


Figura 48. Representação gráfica da quantidade de linhas de código que foram necessárias para desenvolver as aplicações implementadas. A coluna da esquerda representa a aplicação ePaga, enquanto a coluna da direita representa as duas aplicações que contêm cada um dos protocolos em separado. Os blocos divididos verticalmente separam os componentes e respectivas linguagens de programação que formam a aplicação. O bloco JavaME da coluna da direita foi calculado através da soma dos blocos JavaME das duas aplicações correspondentes aos dois protocolos implementados.

Visto que a memória disponível para instalação de código no elemento seguro é um recurso limitado, foi analisado o desempenho da aplicação implementada neste aspecto. Os resultados desta análise estão apresentados na Figura 49. Os valores apresentados mostram que a memória necessária para instalar o *middleware* num elemento seguro é semelhante à ocupada por um protocolo de pagamento. Pela Figura 49 verifica-se que o *middleware* reduz o espaço necessário para desenvolver protocolos, mas o valor desta redução varia significativamente consoante o protocolo. Conclui-se também que, como seria de esperar, esta mais-valia torna-se mais relevante à medida que o número de protocolos instalados aumenta.

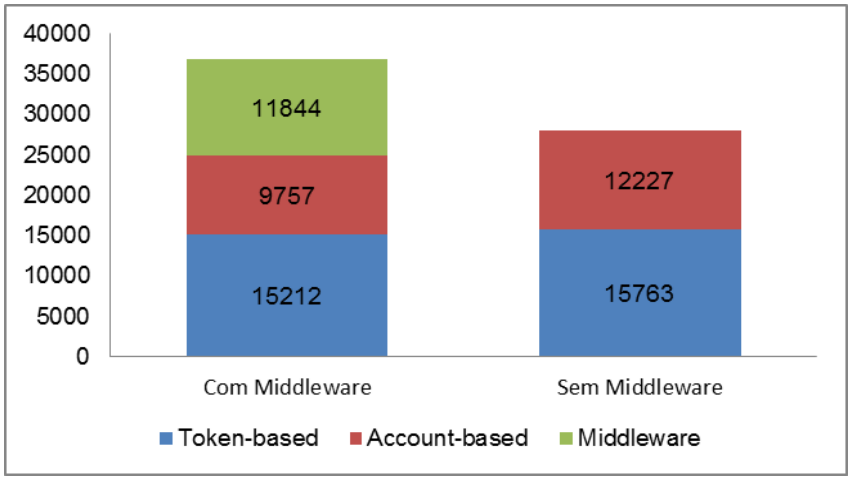


Figura 49. Representação gráfica da memória ocupada pelas aplicações implementadas no elemento seguro. A coluna da direita representa o espaço ocupado pela aplicação ePaga com os dois protocolos instalados. A coluna da direita representa o espaço que ocupariam as duas aplicações dedicadas a cada protocolo, se fossem instaladas no dispositivo. No momento de recolha destes dados, as aplicações encontravam-se no seu estado inicial.

6 Conclusões

Os pagamentos móveis tardam em conquistar definitivamente o mercado. No entanto, a área dos pagamentos móveis continua a evidenciar um potencial tremendo. Para além dos inúmeros sistemas existentes, categorizados e exemplificados neste documento, verifica-se um investimento constante em novos projectos um pouco por todo o mundo [54][55]. Por outro lado, os testes piloto que foram efectuados com este tipo de sistema continuam a comprovar a disposição do público para aderir a um destes serviços [56].

Este número elevado de iniciativas, em conjunto com a actual falta de *standards,* deixa antever um mercado preenchido com sistemas propostos por entidades diferentes, muitos deles incompatíveis entre si. Como resposta a este cenário, foi proposto um sistema que suporta vários protocolos de pagamento. Este sistema permite aos intervenientes de uma transacção comunicarem através do protocolo mais adequado, seleccionado entre os protocolos suportados por ambos.

De seguida apresentam-se as contribuições que a arquitectura do sistema proposto traz ao ecossistema dos pagamentos móveis. Estas contribuições encontram-se resumidas na Tabela 2.

- Reduz o esforço de desenvolvimento e manutenção de novos serviços e protocolos: os protocolos de pagamento actuais estão muitas vezes embutidos nas aplicações que os utilizam, misturando a lógica de negócio do serviço com o pagamento do dito serviço. A arquitectura proposta isola a funcionalidade de pagamento do resto da aplicação. Assim, a implementação de novos serviços é simplificada. Adicionalmente, os protocolos de pagamentos representam módulos separados da restante arquitectura ePaga. Logo, para implementar um novo protocolo de pagamento, deixa de ser necessário implementar uma aplicação de pagamento de raiz. Esta vantagem foi demonstrada no capítulo anterior (Figura 48).
- Aumenta portabilidade dos protocolos desenvolvidos: o mesmo protocolo funciona em qualquer sistema operativo em que a aplicação ePaga esteja implementada.
- Potencia o sucesso de protocolos pouco universais: os sistemas de pagamento existentes evidenciam a dificuldade em criar um protocolo que possa ser utilizado em qualquer situação. No sistema proposto vários protocolos de universalidade limitada formam uma aplicação com uma universalidade superior.
- Maximiza a eficiência do pagamento: mesmo assumindo que existem dois protocolos que podem ser utilizados em qualquer situação, é difícil que um deles seja sempre o mais eficiente. A aplicação ePaga escolhe o protocolo mais eficiente para cada pagamento, reduzindo o seu custo médio.
- Escolha de protocolo automática: como foi explicado anteriormente, soluções como a fornecida pela GlobalPlatform permitem que várias applets convivam no mesmo elemento seguro. Assim, os dois pontos anteriores podiam ser alcançados instalando várias aplicações, cada uma com o seu protocolo. No entanto, a coexistência das aplicações de pagamento não seria transparente para o utilizador. Logo, este teria de escolher o protocolo a usar manualmente, o que se tornaria pouco

prático. Com a aplicação ePaga, a escolha do protocolo é feita de forma automática e transparente para o utilizador.

- Interface constante em função do protocolo escolhido: outra desvantagem do uso de várias aplicações é o acoplamento de uma interface e um protocolo de pagamento. Com o sistema ePaga, a interface passa a estar dependente da acção pretendida pelo utilizador, e não do protocolo que está a ser usado.

Tabela 2. Comparação do sistema ePaga com as alternativas existentes. A primeira coluna representa uma aplicação de pagamento com um protocolo. A segunda coluna representa a utilização de várias aplicações de pagamento, cada uma com um protocolo.

	Uma aplicação	Várias aplicações	ePaga
Gestão de Ciclo de vida	Intermédia	Complexa	*Midlets*: simples *Applets*: complexa
Gestão de Ciclo de vida com GP	Simples	*Midlets*: complexa *Applets*: simples	Simples
Custo de novos protocolos	Alto	Alto	Baixo
Complexidade de uso	Baixa	Alta	Baixa
Eficiência de pagamento	Baixa	Intermédia	Alta
Universalidade necessária em cada protocolo	Alta	Baixa	Baixa

Adicionalmente apresentam-se as seguintes conclusões retiradas da avaliação efectuada e que derivam dos requisitos dos sistemas de pagamento.

- O sistema não acrescenta um *overhead* significativo ao custo por transacção: poderia ser aumentado o custo se o sistema elevasse a quantidade de dados trocada remotamente durante a transacção.
- O sistema aumenta o tempo de uma transacção de proximidade: o valor do *overhead* terá de ser determinado a partir de testes num ambiente mais próximo do real. Ainda assim, no exemplo usado na avaliação, o protocolo de pagamento continuaria a cumprir os requisitos a que se propôs (abaixo de 15 s).

Com estas contribuições, o sistema ePaga, cujo conceito foi demonstrado nesta tese, visa ajudar a explorar o verdadeiro potencial dos pagamentos móveis.

6.1 Trabalho Futuro

Nesta secção enumeram-se os passos que se devem seguir à implementação da prova de conceito descrita neste documento.

- Testes num ambiente mais próximo do real, nomeadamente com dispositivos móveis reais e utilizadores. Estes testes oferecem uma noção mais precisa da performance do sistema.

- Implementação de protótipos em outros sistemas operativos. Este passo explora a portabilidade que a arquitectura proposta permite.
- Implementar o pedido de serviço automático descrito na arquitectura. A interacção com páginas web de vendedores e etiquetas NFC simplifica e acelera o processo de pagamento.
- Adicionar mais protocolos ao sistema. A inclusão de novos protocolos ao sistema permite detectar pontos a melhorar na arquitectura ou na implementação. Quanto mais díspares forem os protocolos, melhor se demostra a versatilidade do sistema.

Referências

[1] Mark Weiser, Some computer science issues in ubiquitous computing, *Communications, ACM* 36, 7 (Julho de 1993), 75-84, 1993

[2] Global mobile cellular subscriptions, total and per 100 inhabitants, 2000-2010. [Online] http://www.itu.int/ITU-D/ict/statistics/material/graphs/2010/Global_mobile_cellular_00-10.jpg

[3] Andriew S. Lim, Inter-consortia battles in mobile payments standardisation, Electronic Commerce Research and Applications, Volume 7, Issue 2, Special Section: Research Advances for the Mobile Payments Arena, Summer 2008, Pages 202-213, ISSN 1567-4223

[4] Karnouskos, S. Mobile payment: A journey through existing procedures and standardization initiatives, Communications Surveys & Tutorials, IEEE, vol.6, no.4, pp.44-66, Fourth Quarter 2004

[5] N. Kreyer, K. Pousttchi, and K. Turowski, Characteristics of Mobile Payment Procedures, Proc. ISMIS 2002 Wksp. m-services, Lyon 2002

[6] M. Hassinen, K. Hyppönen, and K. Haataja, An open, PKI-based mobile payment system, in Proceedings Lecture Notes Computer Science, vol. 3995. pp. 86-100, 2006

[7] Marko Hassinen, Konstantin Hypponen, Elena Trichina, Utilizing national public-key infrastructure in mobile payment systems, Electronic Commerce Research and Applications, Volume 7, Issue 2, Special Section: Research Advances for the Mobile Payments Arena, Summer 2008, Pages 214-231, ISSN 1567-4223

[8] European Payments Council, SEPA Credit Transfer Scheme Rulebook, Novembro 2010

[9] European Payments Council, SEPA Core Direct Debit Scheme Rulebook, Novembro 2010

[10] European Payments Council, SEPA Informação para o Sector Público, Abril 2010

[11] Extensible Markup Language (XML) 1.0 (Fifth Edition) [Online] http://www.w3.org/TR/xml/

[12] European Payments Council, EPC e-Mandates e-Operating Model Detailed Specification, Abril 2009

[13] European Payments Council, White Paper on Mobile Payments v2.0, Junho 2010

[14] EPC, GSMA, Mobile Contactless Payments Service Management Roles - Requirements and Specifications, Outubro 2010

[15] GlobalPlatform, GlobalPlatform's Proposition for NFC Mobile: Secure Element Management and Messaging, Abril 2009

[16] S. Karnouskos et al., Secure Mobile Payment — Architecture and Business Model of SEMOPS, EURESCOM Summit 2003, Evolution of Broadband Service, Satisfying User and Market Needs, 29 Setembro–1 Outubro 2003, Heidelberg, Germany.

[17] Xiaolin Zheng, Deren Chen, Study of Mobile Payments System, E-Commerce Technology, IEEE International Conference on, p. 24, 2003 IEEE International Conference on E-Commerce Technology (CEC'03), 2003

[18] M. Naor and M. Yung, Universal one-way hash functions and their cryptographic applications, In Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC '89), D. S. Johnson (Ed.), ACM, New York, NY, USA, 33-43, 1989

[19] T. S. Fun, L. Y. Beng, J. Likoh and R. Roslan. A Lightweight and Private Mobile Payment Protocol by Using Mobile Network Operator. Proceedings of International Conference on Computer and Communication Engineering (ICCCE 2008), pp. 162–166, 2008

[20] CGD - Cartões Pré-Pagos. [Online] http://www.cgd.pt/Particulares/Gerir-Dia-a-Dia/ Solucoes-Jovens/Cartoes-Jovens/Cartoes-PrePagos/Pages/Cartoes-PrePagos.aspx

[21] Xiaosong Hou, Chik How Tan, A new electronic cash model, Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on , vol.1, pp. 374- 379 Vol. 1, 4-6 Abril 2005

[22] Rahul M. Godbole and Alwyn R. Pais, Secure and efficient protocol for mobile payments, In Proceedings of the 10th international conference on Electronic commerce (ICEC '08), ACM, New York, NY, USA , Article 25 , 10 pages, 2008

[23] R. K. Balan, N. Ramasubbu, K. Prakobphol, N. Christin, and J. Hong, mFerio: The design and evaluation of a peer-to-peer mobile payment system. In Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys), Krakow, Poland, Junho 2009

[24] H. Kreft, Cashing up with Mobile Money – the fairCASH Way, Euro mGov 2005, Sussex University, Brighton, United Kingdom, 10-12 Julho 2005

[25] Jos Dumortier et al. The Legal and Market Aspects of Electronic Signatures. Study for the European Commission

[26] Kungpisdan S., Srinivasan B., and Phu Dung Le, A Secure Account-based Mobile Payment Protocol, Proceedings of the International Conference on Information Technology: Coding and Computing, Vol. 1, Las Vegas, USA, 2004, pp. 35-39

[27] Mahil Carr, Mobile Payment Systems and Services: An introduction, Mobile Payment Forum, 2007

[28] GSM World. Market Data Summary. [Online]

http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm

[29] Saleem Kadhiwal, Anwar Usman Shaheed Zulfiquar, Analysis of mobile payment security measures and different standards, Computer Fraud & Security, Volume 2007, Issue 6, Junho 2007, Pages 12-16, ISSN 1361-3723

[30] In U.S., SMS Text Messaging Tops Mobile Phone Calling. [Online]

http://blog.nielsen.com/nielsenwire/online_mobile/in-us-text-messaging-tops-mobile-phone-calling/

[31] SMS Continues to Confound Expectations as Worldwide Messaging Revenues set to exceed USD 233 billion by 2014. [Online] http://www.portioresearch.com/MMF10-14_press.html

[32] ETSI GPRS (General Packet Radio Service) technology page [Online]

http://www.etsi.org/website/technologies/gprs.aspx

[33] ETSI Universal Mobile Telecommunications System (UMTS) technology page [Online] http://www.etsi.org/WebSite/Technologies/UMTS.aspx

[34] 3GPP – HSPA [Online] http://www.3gpp.org/HSPA

[35] 3GPP – LTE [Online] http://www.3gpp.org/LTE

[36] Mobile WiMAX-Part I: A Technical Overview and Performance Evaluation, WiMAX Forum White paper, June 2006

[37] IRDC, Part 1: Physical Layer. Vishay Semiconductors.

http://irda.org/associations/2494/files/Publications/Physical Basics.pdf

[38] J.J. Chen, C. Adams, Short-range wireless technologies with mobile payments systems, Proceedings of the Sixth International Conference on Electronic Commerce (ICEC), Delft, The Netherlands, October 25–27, ACM International Conference Proceeding Series, vol. 60, ACM Press, New York, NY, USA, 2004

[39] Bluetooth SIG, Bluetooth Specification Version 2.1 + EDR, 2007

[40] D. Sharmila, R. Neelaveni, K. Kiruba, Bluetooth Man-In-The-Middle attack based on Secure Simple Pairing using Out Of Band association model, Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009. 2009 International Conference on , pp.1-6, 4-6 Junho 2009

[41] RFID Journal: Frequently Asked Questions [Online] http://www.rfidjournal.com/faq

[42] Diogo Simões, Sistema de Fidelização sobre NFC (Near Field Communication), Setembro 2008

[43] Information technology - Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1), ISO/IEC 18092, First Edition, Abril de 2004

[44] Multiple NFC-enabled Android handsets to arrive from late 2010 [Online]

http://www.nearfieldcommunicationsworld.com/2010/04/23/33506/multiple-nfc-enabled-android-handsets-to-arrive-from-late-2010/

[45] One in five mobile phones to feature NFC by 2012 [Online]

http://www.nearfieldcommunicationsworld.com/2010/06/04/33823/one-in-five-mobile-phones-to-feature-nfc-by-2012/

[46] A. Ramfos, S. Karnouskos, A. Vilmos, B. Csik, P. Hoepner, and N. Venetakis, SEMOPS: Paying with Mobile Personal Devices, Fourth IFIP Conference on e-Commerce, e-Business, and e-Government(I3E), Toulouse, France, 22-27 August 2004

[47] Ling Zhang, Jianping Yin, Mengjun Li, A Novel Off-line Anonymous and Divisible Digital Cash Protocol Utilizing Smart Card for Mobile Payment, Communications and Networking in China, 2006, ChinaCom '06, First International Conference on, pp.1-6, 25-27 Oct. 2006

[48] C. Schnorr, Efficient Signature Generation by Smart Cards, Cryptology, vol. 4, pp. 161-174, 1991

[49] D. Chaum, Blind Signature for Untraceable Payments, Proc. Advances in Cryptology–Crypto ’82, pp. 199-203, 1983

[50] David Chaum and Eugène Van Heyst, Group signatures, In Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques (EUROCRYPT'91), Donald W. Davies (Ed.), Springer-Verlag, Berlin, Heidelberg, pp. 257-265, 1991

[51] SIBS [Online] http://www.sibs.pt/

[52] SIBS: MB PHONE [Online] http://www.sibs.pt/pt/mb/prodserv/mbphone/

[53] CGD Mobile Payments [Online]
http://www.movensis.com/2010/index.php/bankingpayments/ casosprat/cgd-mobile-payments

[54] O2 Wallet launches in the UK: NFC to be added “in time” [Online]
http://www.nfcworld.com/2012/04/26/315328/o2-wallet-launches-in-the-uk-nfc-to-be-added-in-time/

[55] Samsung partners with FeliCa for Japanese NFC solutions, unveils 2012 Olympics' mobile payment app with Visa [Online]
http://www.engadget.com/2012/02/24/samsung-partners-with-felica-for-japanese-nfc-solutions-unveils/

[56] “la Caixa”, Telefonica and Visa successfully complete the first mobile payments experience in Spain [Online] http://www.mobeyforum.org/Press-Documents/Industry-News/ la-Caixa-Telefonica-and-Visa-successfully-complete-the-first-mobile-payments-experience-in-Spain