

Capítulo

3

Sistemas de Realidade Aumentada Móvel Suportados por Computação em Nuvem

Calebe de Paula Bianchini e Luciano Silva

Laboratório de Processamento Gráfico e Mídias Digitais, Universidade Mackenzie

Laboratório de Sistemas Distribuídos, Universidade Mackenzie

Abstract

Augmented Reality Systems (AR) that allow mobility of sensing systems (cameras, GPS, accelerometers) are called AR mobile systems. Even with the growth of computational power and storage of mobile devices, many AR operations still need to be performed in systems with higher processing and storage capacities. Thus, this chapter aims to present the basis of cloud computing, via cloudlets, as a feasible environment for development of mobile AR applications that require distribution processing.

Keywords: *cloudlets, Mobile AR, AR, Cloud Computing, Distributed Systems.*

Resumo

Sistemas de Realidade Aumentada (RA) que permitem mobilidade dos sistemas de sensoriamento (câmeras, GPS, acelerômetros) são chamados de sistemas de RA móvel. Mesmo com o crescimento do poder computacional e de armazenamento dos dispositivos móveis, muitas operações de RA ainda precisam ser efetuadas em sistemas com maiores capacidades de processamento e armazenamento. Assim, este capítulo tem como objetivo apresentar as bases da Computação em Nuvem, via cloudlets, como ambiente viável para desenvolvimento de aplicações de RA móvel que necessitem de distribuição de processamento.

Palavras-chave: *cloudlets, RA móvel, RA, Computação em Nuvem, Sistemas Distribuídos.*

3.1 Introdução

Sistemas de RA móvel permitem a mobilidade dos seus sensores. Muitas vezes, mesmo com esta mobilidade e devido ao poder de processamento e armazenamento dos dispositivos que dão suporte aos sensores, ainda é necessário realizar operações mais complexas em estações com maiores capacidades de processamento e armazenamento. Além disto, espera-se que estes serviços extra-dispositivo estejam disponíveis de forma pervasiva, ou seja, em todo lugar.

Dentro deste contexto, a Computação em Nuvem apresenta-se como um ambiente muito viável para suporte à distribuição e pervasividade de processamento e armazenamento. Neste ambiente, processamento e armazenamento essenciais ficam vinculados aos dispositivos móveis e, tarefas mais complexas, em nuvens locais e públicas.

Na Computação em Nuvem, existe uma forma de ser organizar o acesso aos serviços por uma arquitetura conhecida como *cloudlet*. Numa *cloudlet*, tem-se um agente de integração, que disponibiliza facilidades de comunicação e invocação de serviços. Recentemente, esta forma de organização permitiu levar aplicações de RA, com necessidades de distribuição de processamento e armazenamento, para nuvens.

Com o objetivo de introduzir o tema a alunos, professores e pesquisadores de RA, este capítulo apresenta as bases essenciais para integração de aplicações de RA com Computação em Nuvem, especificamente sob a arquitetura de *cloudlets*. A apresentação está organizada da seguinte forma:

- inicialmente, são apresentados os principais conceitos de Computação em Nuvem, mostrando as suas várias formas de implementação;
- em seguida, são revisados os conceitos fundamentais de RA e são apresentados os fundamentos de RA móvel;
- finalmente, através de um exemplo concreto, é mostrada como é possível a integração dispositivos móveis com sistemas de nuvem, tendo como mediador uma *cloudlet*.

Ao final do texto, são sugeridas algumas referencias bibliográficas que aprofundam os temas tratados neste capítulo.

3.2 Fundamentos em Computação em Nuvem

3.2.1 Modelo de Computação em Nuvem

O modelo de Computação em Nuvem, como é definido e apresentado nos dias atuais, só foi possível devido à evolução natural da Computação e da Internet. Na verdade, porém, esse conceito remonta os primórdios da própria Internet, quando Leonard Kleinrock, da ARPANET, comentou que um modelo de computação utilitária (*utility computing*) surgiria com o amadurecimento e sofisticação das redes de computadores de forma

semelhante às redes de água, de energia elétrica, de telefone, dentre outras commodities já consolidadas e de uso pessoal (Buyya 2013).

Dessa forma, na Computação em Nuvem, os usuários podem fazer uso dos serviços disponíveis sem precisarem considerar a localização de onde esses provedores de serviços se encontram. Além disso, um dos princípios desse modelo considera que o usuário paga apenas pelo serviço consumido, ou seja, *pay-per-use*.

Considerando que esse modelo está influenciando toda a Computação, fica nítido perceber seus resultados no projeto de sistemas computacionais, no desenvolvimento de aplicações, na entrega de soluções e na execução de softwares. De forma mais geral, todas as áreas que são regidas por qualquer tipo de serviço, não somente software, mas também capacidade de processamento, *storage*, redes de computadores, e até mesmo infraestrutura, podem ser construídas e oferecidas como Computação em Nuvem. A Figura 1 apresenta uma visão geral da Computação em Nuvem, representando a interconectividade entre diversos dispositivos.

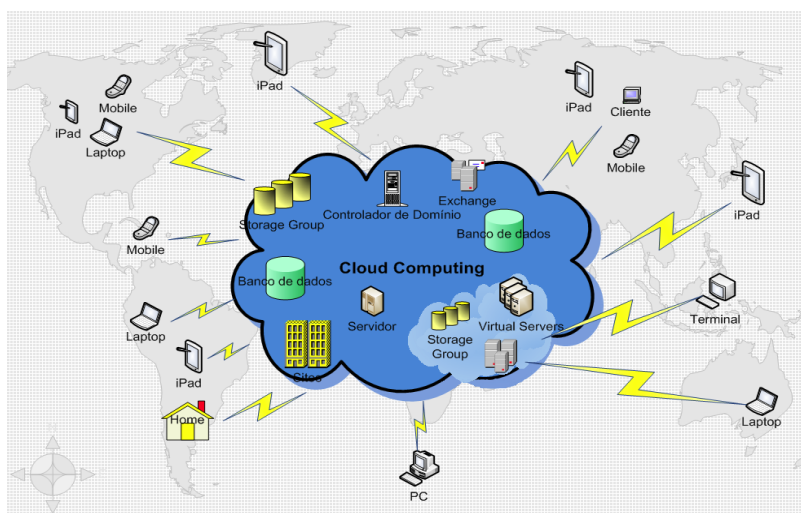


Figura 1. Visão geral da Computação em Nuvem.
Fonte: (Lisboa 2014).

Em função da possibilidade de abstração que o termo Computação em Nuvem possui, perceptível na imagem da Figura 1, esse termo tem sido usado para diversas situações no mundo de TI, permitindo uma mistura de tecnologias, serviços e conceitos. Apesar do símbolo de uma Nuvem já ser utilizado há um bom tempo para representar uma rede de computadores, na Computação em Nuvem tem um forte apelo à centralidade da Internet.

Assim, diversas definições têm sido utilizadas para a Computação em Nuvem, sendo que algumas das principais são apresentadas a seguir:

- Segundo (NIST 2011), “a computação em nuvem é um modelo que permite uma rede de acesso, de forma ubíqua e conveniente, à um conjunto de recursos computacionais (como, por exemplo, redes de computadores, servidores,

storages, aplicações e serviços) que são rapidamente provisionados com o menor esforço possível ou a menor quantidade de interação com um provedor.”¹

- Segundo (Hwang 2011), “computação em nuvem é um paradigma de computação de alta vazão (*high-throughput*) em que uma infraestrutura fornece diversos serviços por meio de grandes datacenters. Esse modelo permite que usuários compartilhem recursos de qualquer lugar, a qualquer momento, por meio de dispositivos interconectados.”²
- Segundo (Armbrust 2009), “computação em nuvem se refere tanto aos aplicativos disponíveis como serviço na Internet quanto o hardware e a infraestrutura de software básico nos datacenters que oferecem esse serviço.”³
- Segundo (Gartner 2014): “computação em nuvem é um estilo de computação no qual escalabilidade e elasticidade de recursos de TI são entregues como serviço para clientes externos por meio da Internet.”⁴
- Segundo (Buyya 2013), “computação em nuvem é um tipo de sistema paralelo e distribuído que consiste de uma rede de computadores interconectados e que são dinamicamente provisionados, apresentados como um ou mais recursos computacionais unificados com acordos de nível de serviço estabelecidos entre os clientes e provedores.”⁵
- Segundo (Erl 2013), “computação em nuvem é uma forma especializada de computação distribuída que introduz modelos utilitários para prover recursos remotos, de forma escalável e mensurável”.⁶

¹ Tradução literal da definição apresentada em (NIST 2011): “*Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.*”

² Tradução literal da definição apresentada em (Hwang 2011): “*Cloud computing is a high-throughput computing (HTC) paradigm whereby the infrastructure provides the services through a large data center or server farms. The cloud computing model enables users to share access to resources from anywhere at any time through their connected devices.*”

³ Tradução literal da definição apresentada em (Armbrust 2009): “*Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services.*”

⁴ Tradução literal da definição apresentada em (Gartner 2014): “*Gartner defines cloud computing as a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service using Internet technologies.*”

⁵ Tradução literal da definição apresentada em (Buyya 2013): “*A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers.*”

⁶ Tradução literal da definição apresentada em (Erl 2013): “*Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources.*”

É possível perceber que, em função dessa quantidade de definições, o termo Computação em Nuvem ainda não está totalmente estabelecido. Alguns autores tem se esforçado para relatar o estado-da-arte da Computação em Nuvem até então, visando não só encontrar algum ponto em comum nas definições existentes, mas também estabelecer algum critério comparativo entre elas (Vaquero 2008) (Zhang 2010). Pode-se afirmar que, dentre as definições atuais, aquela apresentada pelo NIST (2001) é mais bem aceita pela comunidade. Assim, neste trabalho, a definição do NIST (2011) é mais apropriado para estudo da Computação em Nuvem.

As principais características da Computação em Nuvem são aquelas que refletem exatamente as mudanças que esse novo paradigma traz. Essas características podem ser divididas em cinco grandes categorias:

- **Autosserviço sob demanda:** é a possibilidade de o cliente fazer suas próprias escolhas de recursos computacionais, como servidores e redes, em diversos provedores. Conforme essas escolhas, a Computação em Nuvem é reconfigurada de acordo com sua expectativa e, dessa forma, passa a atender melhor suas necessidades. Além disso, a interação com o cliente se dá de forma transparente e automática, sem a necessidade de intervenção humana.
- **Extenso acesso por rede:** os recursos computacionais utilizados na Computação em Nuvem podem ser acessados e configurados por meio de protocolos e mecanismos padronizados, por meio dos diversos tipos de dispositivos, como *tablets*, celulares e computadores. Questões de heterogeneidade não devem ser percebidas no acesso a esses recursos computacionais, ou seja, plataformas *thin* ou *thick client*, linguagens de programação e/ou sistema operacionais não devem interferir no acesso e na qualidade.
- **Agrupamento de recursos:** os provedores de recursos computacionais para a Computação em Nuvem devem atender diversos clientes (*mult-tenant model*) gerenciando os recursos físicos ou virtuais dinamicamente para que possam ser alocados e realocados conforme a demanda do cliente. Os clientes também não devem ter a exata noção de localização dos recursos, mas possuir um grau natural de abstração, como estado, país, datacenter, etc.
- **Elasticidade:** os recursos podem aumentar e diminuir elasticamente. Em alguns casos, essa elasticidade pode ser feita de forma automática, visando um aumento rápido na escalabilidade, ou uma diminuição, conforme demanda. Para os clientes, a visão de escalabilidade permite visualizar recursos de forma infinita e disponível a qualquer momento. É neste contexto que uma camada abstrata e virtualizada permite diversas formas de elasticidade.
- **Medição de Serviço:** sistemas de Computação em Nuvem devem automaticamente controlar e otimizar o uso dos recursos por meio de técnicas de medição. Essas medidas representam um grau de abstração apropriado tanto para os clientes quanto para os provedores (largura de banda, grau de processamento, tamanho de armazenamento, etc). Assim, os recursos podem ser monitorados, controlados e relatados em um grau de transparência para os dois lados envolvidos: cliente e provedor. SLA (*Service Level Agreement*) é uma forma natural de garantir que os recursos possuem certo grau de qualidade (*QoS*).

Outra forma de analisar a Computação em Nuvem é em relação à implantação da “Nuvem”. Ou seja, quem são os responsáveis pela administração e quais são os limites estabelecidos sobre a infraestrutura permitem tipificar uma Nuvem. Nesse caso é possível analisar como se dá a relação entre as diversas Nuvens, entre os diversos provedores. Sabe-se, porém, que cada provedor possui sua forma única de administrar a infraestrutura e os serviços disponíveis. A Figura 2 apresenta uma junção de tipos diferentes de Nuvem, destacando a relação entre os diversos provedores por meio da Internet.

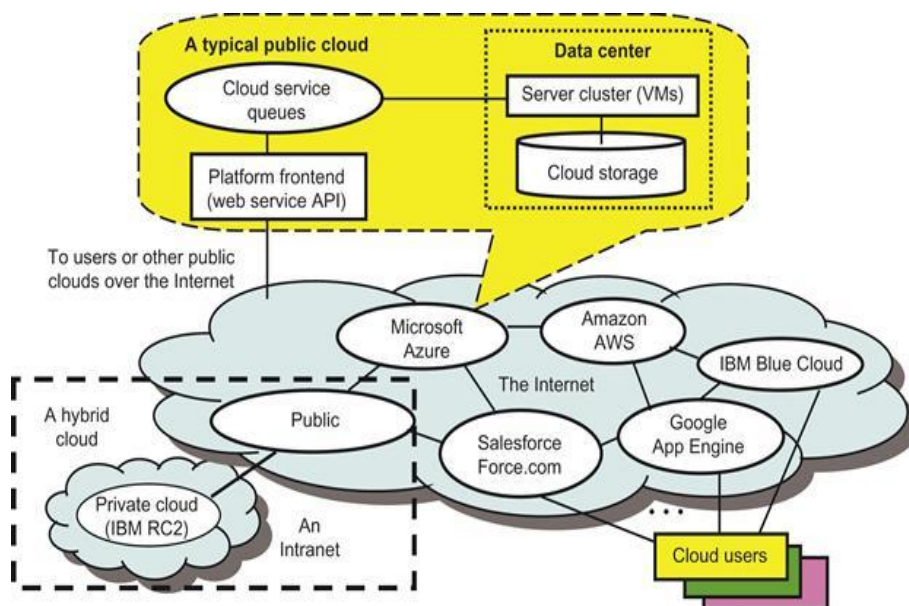


Figura 2. Limites administrativos: nuvem pública, privada e híbrida.
 Fonte: (Hwang 2011)

De forma geral, os tipos de Nuvem são:

- **Pública:** esta Nuvem está disponível para uso do público em geral. A Nuvem pode pertencer a empresas, governos ou universidade, ou uma combinação deles, que gerenciam e disponibilizam os recursos e serviços para serem utilizados por qualquer usuário. Cada provedor de Nuvem Pública possui sua política administrativa e de garantias de serviço, e pode oferecer mecanismos diferenciados de *pay-per-use*. Neste caso, é muito comum o uso de padrões já estabelecidos de mercado para o oferecimento de recursos e serviços.
- **Privada:** uma Nuvem privada só está disponível para uso de uma única empresa, sendo esta responsável pela administração, gerenciamento e disponibilização dos serviços. Normalmente é implantando em uma intranet (mas não é exclusivo), e permite que seus usuários tenham acesso a uma Nuvem de forma rápida e exclusiva. Nesse caso, imagina-se que uma Nuvem Privada ofereça serviços mais eficientes, ágeis, úteis e simples, com o custo de utilizar padrões específicos da própria empresa. Da mesma forma, outros mecanismos de faturamento sobre os serviços podem ser utilizados.

- **Comunitária:** uma Nuvem comunitária considera que, dado algum interesse em comum, como missão, segurança, requisitos, valores, etc., usuários específicos de diversas organizações façam uso de seus recursos e serviços. Nesse caso, esta Nuvem pode ser proprietária a uma ou mais organizações, sendo estas responsáveis pelo seu gerenciamento, controle e administração. Por envolver diversas organizações, as políticas administrativas seguem o interesse comum estabelecido e cada organização poderá oferecer recursos e serviços que contribuam para a Nuvem Comunitária. Neste caso, mecanismos específicos de faturamento devem ser estabelecidos para que a Nuvem não tenha seus recursos e serviços comprometidos.
- **Híbrida:** esta Nuvem é formada pela combinação de, pelo menos, duas Nuvens dos outros tipos. Apesar dessa integração, cada Nuvem ainda é única em sua essência e mantém suas propriedades. Elas são integradas por mecanismos padronizados ou proprietários que permite portabilidade de dados, serviços e recursos. Nessa implantação, as políticas de gerenciamento, controle e administração podem ser utilizadas para que seja escolhido o que há de melhor cada tipo de Nuvem.

Uma Nuvem deve ser organizada com uma pilha de hardwares e softwares caracterizando sua arquitetura. Cada arquitetura define os serviços que podem ser oferecidos por uma Nuvem. A Figura 3 apresenta a organização em pilha de uma Nuvem com seus possíveis tipos de serviços.

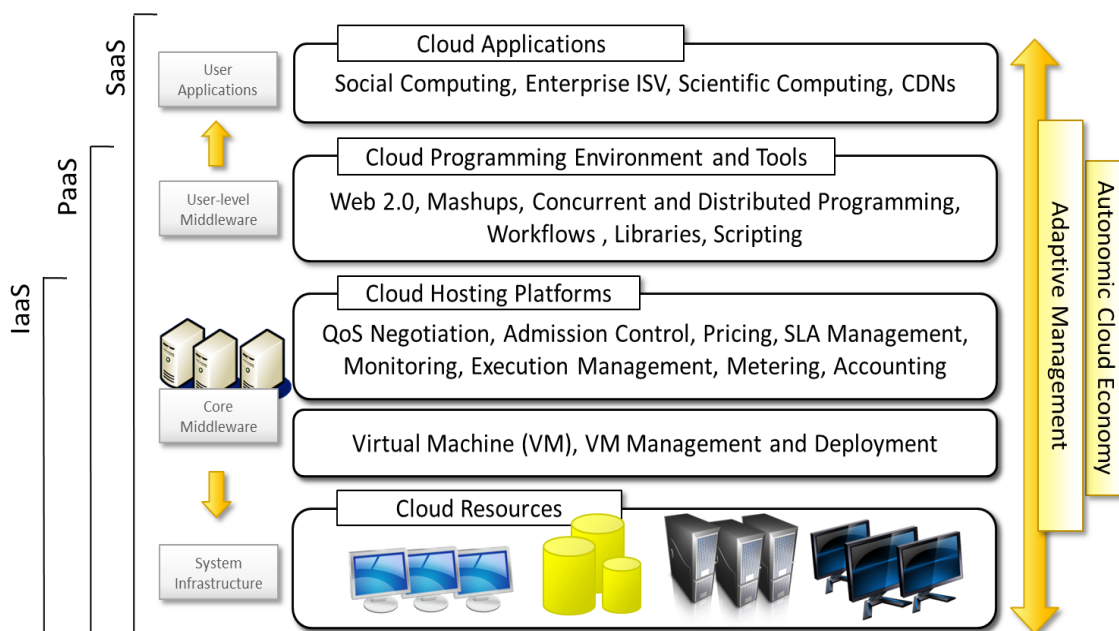


Figura 3. Arquitetura de uma Nuvem.
 Fonte: Adaptado de (Buyya 2013)

A primeira camada, *Cloud Resources*, representa os recursos iniciais necessários para uma Nuvem: a infraestrutura física de hardware. A próxima, *Cloud Hosting*

Platforms, é um conjunto de softwares de administração, gerenciamento e controle para as aplicações que devem funcionar nos melhores recursos disponíveis. Virtualização e máquinas virtuais são as técnicas mais utilizadas nessa camada. A combinação destas duas camadas em uma arquitetura é chamada de *Infrastructure-as-a-Service* (IaaS).

A próxima camada, *Cloud Programming Environment and Tools*, oferece ambientes e ferramentas para o desenvolvimento de serviços para uma Nuvem. Assim, usuários podem desenvolver suas próprias aplicações utilizando APIs e middlewares específicas. Por isso, a arquitetura que utiliza essa camada é conhecida como *Platform-as-a-Service* (PaaS).

Por fim, a última camada, *Cloud Applications*, representa todos os serviços e aplicações disponíveis para usuários. Normalmente, nesta camada estão as aplicações para *web*. Esta arquitetura é melhor conhecida como *Software-as-a-Service* (SaaS).

Além dessas definições de arquitetura, alguns outros autores ainda apresentam outras categorias, como pode ser observado em (Youseff 2008) e (Linthicum 2014).

3.2.2 Infrastructure-as-a-Service (IaaS)

A intenção de um IaaS é fornecer recursos de processamento, *storage*, rede, ou qualquer outro recurso computacional de hardware ou infraestrutura física de tal forma que um consumidor (ou usuário) possa utilizar esses recursos para implantar qualquer tipo de software, incluindo sistemas operacionais ou aplicações.

Esse é o formato mais comum da Computação em Nuvem. Os provedores conseguem entregar as mais diversas e complexas configurações desses recursos. Para isso, virtualização é a tecnologia mais utilizada.

Virtualização é um termo guarda-chuva que representa um grau de abstração de um ambiente computacional. A intenção principal da virtualização é permitir o compartilhamento de recursos computacionais, como CPU, disco memória, ou, até mesmo, bibliotecas e softwares. Além do compartilhamento, existem outras principais características de ambientes virtualizados, como agregação, emulador e isolamento. No caso de IaaS, virtualização normalmente é utilizado para representar a abstração de um hardware.

Máquinas virtuais (VM – *virtual machines*) são as unidades básicas mais comum de um provedor de IaaS. É possível configurar quantidade de processadores, tamanho de memória e de disco. Ainda é possível utilizar imagens de VM previamente configurada para organizar uma Nuvem: desde VMs com apenas o sistema operacionais, até mais elaboradas com servidores web e LAMP⁷. As soluções utilizadas para essa camada de virtualização permitem balanceamento, isolamento, área de segurança restrita e ajustes nas VMs.

Na perspectiva do consumidor (o usuário), há um melhor aproveitamento da infraestrutura, diminuição de custos de manutenção e administração e no capital investido em infraestrutura.

⁷ LAMP é um acrônimo para Linux-Apache-MySQL-PHP, uma infraestrutura preparada para executar aplicações em PHP em um servidor Apache, com banco de dados MySQL, em um sistema operacional Linux.

A Figura 4 descreve os componentes básicos de uma infraestrutura de IaaS. No topo, a camada oferece serviços por meio de interfaces gráficas (normalmente, Web 2.0) para o usuário. Nesta camada também pode estar serviços que, por meio de uma API, outras aplicações podem se integrar com o provedor de IaaS.

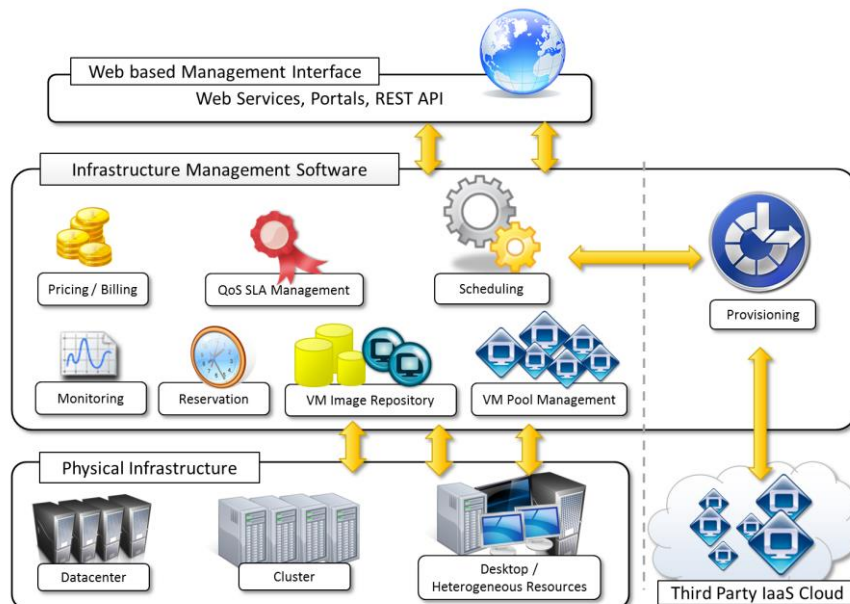


Figura 4. Modelo de referência de um IaaS.

Fonte: (Buyya 2013)

Na camada central, estão disponíveis as principais características de uma solução IaaS. É nesta camada que o gerenciamento de VMs, principal recurso desta arquitetura, é realizado por meio do componente de escalonamento. Este, por sua vez, aloca a execução das VMs nos recursos disponíveis e interage com outros componentes dessa camada para a combinação de novas tarefas: contabilização e precificação, monitoramento, reserva, QoS/SLA, gerenciamento de imagens, gerenciamento de VMs ativas, e interação com componentes de terceiros.

Na camada mais baixa está a infraestrutura física, que podem ser totalmente heterogênea. É esta camada que deve ser construída para atender as necessidades e objetivos da Nuvem. Assim, essa infraestrutura pode ser formada por servidores, cluster, supercomputadores ou todos esses agregados.

Os fornecedores de IaaS mais conhecidos são *Amazon*, *GoGrid*, *Joyent*, *Rightscale*, *Terremark*, *Rackspace*, *ElasticHosts*, e *Flexiscale*.

3.2.3 Platform-as-a-Service (PaaS)

Um provedor de PaaS deve permitir que um consumidor (ou usuário) seja capaz de implantar nessa arquitetura uma aplicação adquirida ou criada em linguagens de programação, bibliotecas, plataformas ou ferramentas compatíveis com o provedor. Esse usuário não controla ou gerencia as camadas inferiores da Nuvem, como rede, servidores, sistemas operacionais ou *storages*, mas tem controle sobre a aplicação e possíveis configurações da aplicação.

A Figura 5 apresenta as principais características de uma arquitetura PaaS. A camada intermediária possui as principais funcionalidades que basicamente está centrada no ambiente de execução para as aplicações. Esta mesma camada não deve expor qualquer característica ou serviço da camada inferior, ou seja, da infraestrutura, para qualquer aplicação implantada. Dessa forma, a infraestrutura pode ser física ou uma arquitetura IaaS, proprietária ou não.

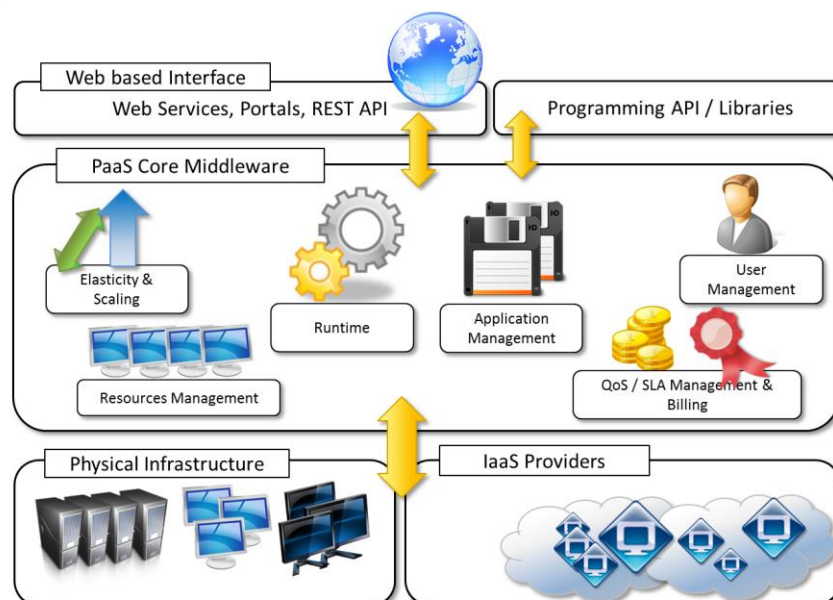


Figura 5. Modelo de referência de um PaaS.
Fonte: (Buyya 2013)

É nesta camada intermediária que estão os processos automatizados de implantação real da aplicação para a infraestrutura, bem como do gerenciamento do balanceamento de carga e controle das bases de dados. Além disso, deve gerenciar a aplicação com base nas políticas estabelecidas pelo próprio usuário (configurações da aplicação). Também é responsabilidade desta camada gerenciar e escalar os recursos alocados para a aplicação sob demanda, possivelmente automaticamente, de acordo com as necessidades do usuário.

Um provedor de PaaS deve oferecer interfaces de interação para o usuário, tanto para o desenvolvimento da aplicação quanto para sua implantação. Essa interface de interação pode ser disponibilizada por meio de interface web, ou até mesmo por meio de linguagens de programação, APIs e bibliotecas.

Além disso, a aplicação desenvolvida para uma PaaS específica segue o modelo de desenvolvimento definidos pelo provedor. Algumas delas utilizam ambiente para linguagens de 4ª geração, programação visual ou ambiente de prototipação rápida (usando componentes previamente disponíveis no ambiente). Outros provedores oferecem uma representação orientada a objetos e APIs específicas para a criação de aplicações. Dessa forma, os desenvolvedores tem uma linguagem de programação clássica, como Java, Python, C#, com algumas restrições para garantir escalabilidade e segurança.

De forma geral, uma arquitetura PaaS deve oferecer:

- Framework de execução: que garante a execução da aplicação do usuário;
- Abstração: permitindo a implantação e gerenciamento da aplicações na Nuvem;
- Automatização: implantar, gerenciar, e escalar a aplicação na infraestrutura física;
- Serviços de Nuvem: fornecer serviços e APIs, ou similares, facilitando a criação de aplicações elásticas e escaláveis.

Nesse sentido, existem provedores totalmente proprietários, como o *Force.com*; provedores que construíram sua arquitetura PaaS sobre padrões da indústria, como o *Google AppEngine* e o *Microsoft Azure*, mas ainda assim permanecendo parte da solução proprietária; e provedores que seguiram padrões abertos, facilitando a migração das aplicações, como *Appistry Cloud IQ Platform*, *Heroku*, e *Engine Yard*.

Além destes, existem também outros provedores conhecidos de PaaS, como o *Longjump*, *AppSale*, *Joyent Smart*, *Platform*, *GigaSpaces XAP*, *DataSynapse*, *Manjrasoft Aneka*, *Apprenda*, *SaaSGrid* e *DataGrid*.

3.2.4 Software-as-a-Service (SaaS)

Um provedor de SaaS deve ser capaz de fornecer softwares deste próprio provedor, executados em uma Nuvem, para um consumidor (ou usuário). Esse software pode ser acessado pelos mais diversos dispositivos e clientes, como celulares e browsers, dentre outros. O usuário, por sua vez, não controla a infraestrutura da Nuvem, nem mesmo a aplicação ou sua execução. No máximo, poderia ter acesso a alguns ajustes de seu perfil ou configurações pessoais.

O conceito de SaaS não é tão recente assim, surgindo na década de 90. Sua aceitação se consolidou em 2001, quando foi oficialmente padronizado e inicialmente definido. Naquele tempo, suas principais características se assemelham muito ao conceito de hoje:

- O produto é vendido para o consumidor como acesso ao aplicativo;
- O software é gerenciado por uma empresa centralizadora;
- O serviço é vendido como “um-para-muitos”;
- O serviço é entregue conforme regras de contrato, conforme prometido.

De forma geral, SaaS é um modelo de entrega de software que provê acesso a aplicativos por meio da Internet e de uma interface web. Esse modelo é atraente para os usuários, inclusive em função das customizações individuais. É nesse formato, de acesso a um aplicativo por diferentes tipos de usuários, que se consolidou o conceito de

“um-para-muitos”. Exemplos de software de CRM⁸ e de ERP⁹, que são utilizados por diversas empresas, podem ser encontrados nesse modelo.

Outra característica de SaaS é que os usuários não precisam fazer a implantação e/ou instalação do software. O acesso é centralizado em um portal e, dadas as credenciais do usuário, ele utiliza o aplicativo. Dessa forma, mediante contrato de faturamento, o usuário é cobrado conforme o uso dos aplicativos e, dependendo do caso, há possíveis alterações e customizações do software.

Outra consequência desse modelo é a centralização no provedor do software. Os esforços são todos concentrados no gerenciamento e manutenção da infraestrutura da arquitetura SaaS, assim como na manutenção e na atualização do serviço oferecido, otimizando os recursos e distribuindo os custos pelos diversos consumidores. Para o consumidor, esse custo é uma mínima fração da taxa de uso do serviço.

Considerando a pilha de hardware e software para uma arquitetura em Nuvem (apresentada na Figura 3), a camada SaaS está no topo. Na verdade, essa camada não só representa SaaS, mas também o conceito de “tudo-como-serviço” (*Everything-as-a-Service*), ou XaaS. Assim, alguns benefícios dessa arquitetura se destacam:

- Redução de custo do software e TCO são evidentes;
- Melhora no nível do serviço;
- Desenvolvimento rápido;
- Aplicações totalmente reconfiguráveis;
- Integração de aplicações e dados (mesmos que rudimentar);
- Assinatura de serviço e pagamento sob demanda.

É nessa perspectiva que o *SalesForce.com* se tornou o exemplo de sucesso mais conhecido de SaaS. Sobre sua plataforma, o *Force.com*, uma arquitetura PaaS, foi desenvolvido um serviço completo de CRM. Além deste, existem outros serviços disponíveis, como o *AppExchange*, no qual é possível pesquisa, publicar e integrar outros serviços e aplicações aos existentes, tornando o *SalesForce.com* uma ferramenta extensível e personalizável.

De forma semelhante, existem outros provedores de serviços relacionados ao gerenciamento empresarial e de relacionamento ao cliente, como o *NetSuite* e *RigthNow*. Estes dois permitem personalização da aplicação final, de acordo com as necessidades do consumidor/cliente.

⁸ CRM, ou *Customer Relationship Management*, é um software que simplifica o processo de gerenciamento de usuários e ajuda na identificação de estratégias de venda.

⁹ ERP, ou *Enterprise Resource Planning*, é um software que permite uma visualização integrada da empresa na tentativa de facilitar seu gerenciamento e melhorar os fluxos de informação entre a perspectiva funcional e os recursos.

Um outro tipo de SaaS existente são aqueles relacionados à redes sociais. Nesse segmento se destacam o *Facebook* e o *LinkedIn*, os quais permitem a adição de plug-ins e integração completa ao perfil dos usuários desses sistemas.

Por fim, existem também serviços de automação de escritório, como o *Box.net*, *Google Documents*, *Zoho Office* e *Microsoft Office 365*. Estes oferecem aplicativos de edição de texto, planilhas eletrônicas e construção de apresentações totalmente desenvolvidas para a Internet. Dessa forma, esses documentos podem ser gerenciados e modificados a partir de qualquer dispositivo, além de permitir que sejam compartilhados entre diversos usuários.

3.3 Realidade Aumentada Móvel

3.3.1 Sistemas de Realidade Aumentada

Realidade Aumentada (RA) pode ser vista como a integração de informações virtuais (modelos bi e tridimensionais, imagens, áudio, vídeo, dentre outros) a visualizações do mundo real. Azuma (1997) define RA como um sistema que:

- combina elementos virtuais com o ambiente real
- é interativa e tem processamento em tempo real
- é concebida em três dimensões

A Figura 6 ilustra um sistema típico de RA onde, a partir de uma imagem captada por uma câmera, um marcador (símbolo) é substituído por um objeto tridimensional.

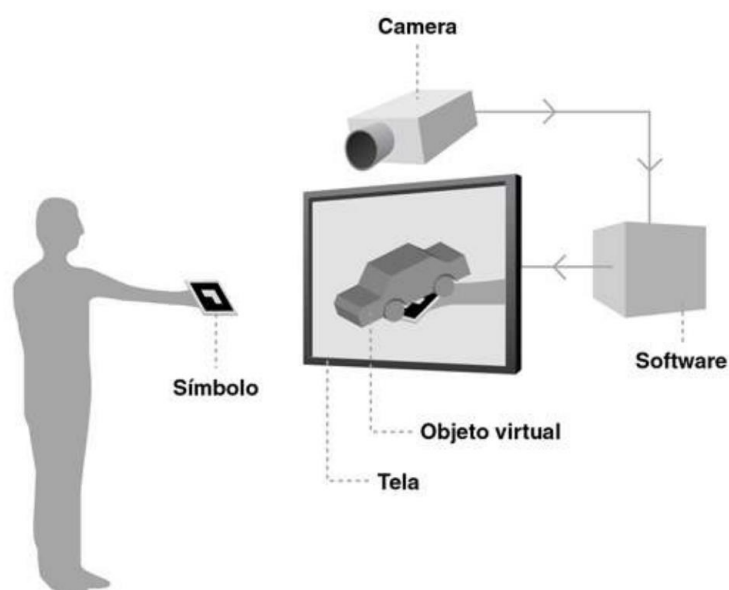


Figura 6. Sistema típico de RA.

Fonte: <http://homes.dcc.ufba.br/~caiosba/mestrado>

Embora existam sistemas de RA não-baseados em marcadores, o uso destes elementos ainda é bastante comum. Existem alguns marcadores bem conhecidos como, por exemplo, o marcador Hiro mostrado na Figura 7:



Figura 7. Marcador Hiro
Fonte: www.artoolworks.com

O processo de substituição do marcador por um elemento virtual é conhecido como fusão. Na Figura 8, tem-se um exemplo de fusão com um modelo tridimensional:

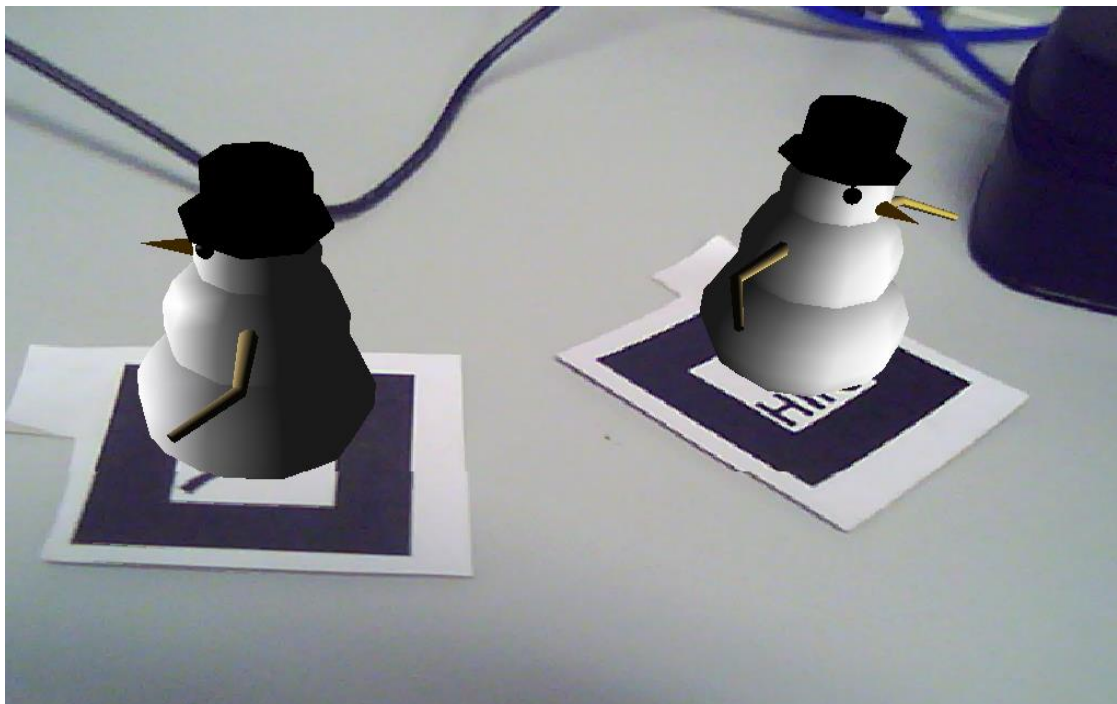


Figura 8. Fusão do marcador Hiro com um objeto tridimensional
Fonte: <http://homes.dcc.ufba.br/~caiosba/mestrado>

Marcadores mais sofisticados podem envolver codificação como, por exemplo, um QR-Code (Figura 9).



Figura 9. QR-Code
Fonte: www.onvert.com

O código presente em um QR-code pode ser convertido em textos interativos, endereços WEB, localizações geo-referenciadas ou mesmo chamadas para programas. A Figura 10 ilustra um exemplo de um QR-code convertido para um post no programa Twitter:



Figura 10. Conversão de um QR-code para uma chamada de programa
Fonte: <http://www.maximumpc.com/files/u46168/toxin20.jpg>

Finalmente, um sistema de RA não precisa utilizar necessariamente um marcador (*markless*). Nesta visão diferenciada, utiliza-se padrões geométricos bem-conhecidos (linhas, polígonos, texturas, dentre outros) como suporte para fusão.

A Figura 11 ilustra um exemplo típico deste tipo de sistema, onde uma textura serviu como base para geração de um mapa digital de elevação:



Figura 11. Geração de modelos digitais de elevação a partir de texturas

Fonte: <http://www.ovelf.com/wp-content/uploads/GIS-map.png>

Principalmente para sistemas de RA móvel, a não-dependência de um marcador é uma característica bastante interessante, pois permite uma exploração maior de elementos reais como suporte para fusão.

3.3.2 Realidade Aumentada Móvel

Sistemas de RA que permitem mobilidade dos sistemas de sensoriamento (câmeras, GPS, acelerômetros) são chamados de sistemas de RA móvel. Atualmente, os sistemas de RA móvel mais representativos estão localizados em *smartphones* e *tablets*, conforme exemplo da Figura 12:



Figura 12. Ambiente de RA móvel instalado em um *smartphone*.

Fonte: <http://www.mysocialagency.com>

Uma das razões do uso crescente de sistemas de RA móvel suporta-se no crescimento do poder computacional (tanto CPU quanto GPU) e de armazenamento de dispositivos móveis como *smartphones* e *tablets*. Além disto, várias bibliotecas e frameworks bem conhecidos como, por exemplo, a ARToolkit já possuem versões próprias para dispositivos móveis.

Além dos *smartphones* e *tablets*, também existem sistemas de AR móveis totalmente dedicados para este fim. Nesta linha, existem sistemas baseados em óculos que só realizam o processo de sensoriamento e recepção das imagens fundidas, conforme exemplo da Figura 13:



Figura 13. Sistema de sensoriamento e recepção de imagens fundidas

Fonte: <http://www.vuzix.com>

Na arquitetura mostrada na figura acima, todo o processamento de fusão dos marcadores é realizado numa estação externa, interligada por redes de comunicação sem fio com o dispositivo. Numa arquitetura contrária, alguns sistemas como o Google Glass, mostrado na Figura 14, já realizam processamentos e fazem armazenamento no próprio dispositivo:



Figura 14. Dispositivo Google Glass

Fonte: <http://www.google.com/glass>

Com o advento do dispositivo Google Glass e a necessidade de se ter acesso móvel pervasivo em vários ambientes, uma arquitetura que permitisse distribuir processamento/armazenamento entre dispositivos móveis e portais de serviços começou a se conformar. Neste contexto, o ambiente de computação em nuvem revelou-se com uma alternativa bastante viável para atender a estes requisitos.

3.3.3 Ambientes de Desenvolvimento para Realidade Aumentada Móvel

Várias bibliotecas e frameworks bem conhecidos em RA convencional já possuem versões para dispositivos móveis.

A biblioteca ARToolkit (Artools 2014), uma das mais conhecidas para desenvolvimento de RA convencional, já possui versões para os dois principais sistemas operacionais móveis: iOS e Android. A Figura 15 mostra exemplos desta biblioteca nos dois sistemas operacionais:



Figura 15. ARToolkit em iOS(esquerda) e Android(direita).

Fonte: <http://www.artoolworks.com>

Ainda para a plataforma Android, existe a biblioteca AndAR (Google Code 2014), disponibilizada sob licença GPL. A Figura 16 mostra um exemplo de uso da AndAR:

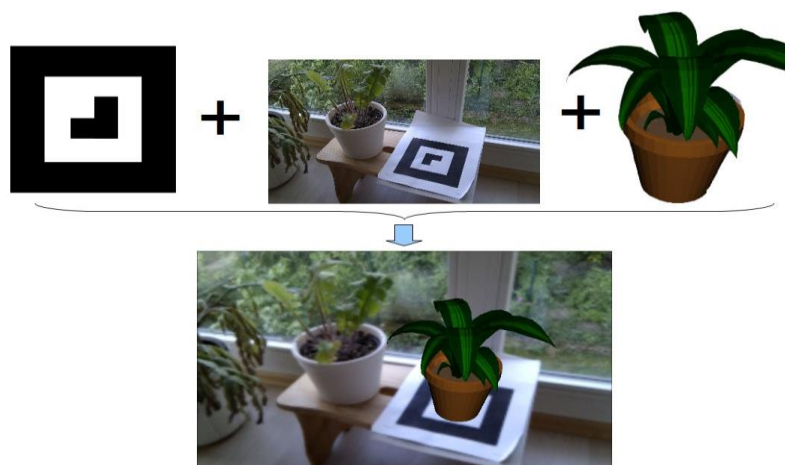


Figura 16. Exemplo de funcionamento da biblioteca AndAR

Fonte: <http://www.artoolworks.com>

Para a plataforma Windows móvel (Windows Phone 8, por exemplo), existe a alternativa da biblioteca GART (Gart 2014). Esta biblioteca permite o desenvolvimento de aplicações de RA que precisam de suporte a georeferenciamento. A Figura 17 mostra o aplicativo Map Bings, que permite a localização de facilidades em uma cidade, construído com base na biblioteca GART:



Figura 17. Aplicativo Map Bings, produzido pela biblioteca GART

Fonte: <http://www.bing.com/maps/>

O uso de todas as bibliotecas mostradas anteriormente requerem acesso via linguagem de programação. A Artoolkit for iOS requer o uso da linguagem Objective-C, a Artoolkit for Android requer Java e, a GART, linguagens da plataforma .NET como C#. Porém, o desenvolvimento de aplicações móveis para RA pode ser simplificado através do uso de ambientes que já geram código para a aplicação. Nesta linha, encontra-se o ambiente Unity3D que, além de permitir o desenvolvimento visual da aplicação, ainda exporta código para diversos tipos de dispositivos móveis.

A Figura 18 mostra a extensão UART (Unity Augmented Reality Toolkit) (Gatech 2014), disponibilizada como um plugin para o Unity3D:

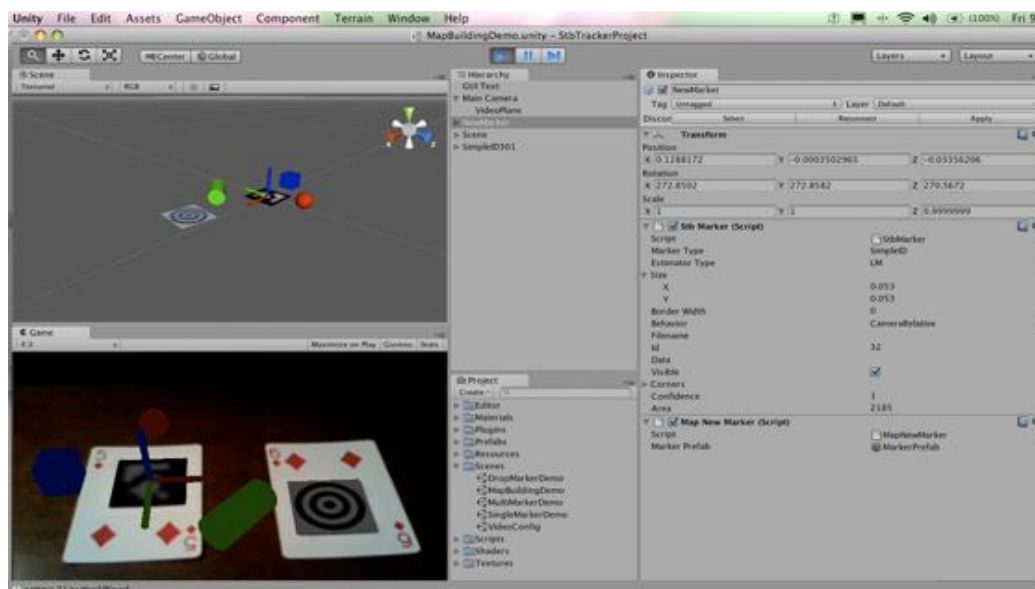


Figura 18. Toolkit UART, integrado ao ambiente Unity3D

Fonte: <https://research.cc.gatech.edu/uart/>

Com o uso do UART, o desenvolvedor da aplicação somente especifica qual o marcador será utilizado e qual será o objeto que substituirá o marcador durante o processo de fusão.

3.4 Integração entre Sistemas de Realidade Aumentada Móvel e Nuvem

3.4.1 Conceito de Cloudlet

Uma *cloudlet* é um elemento arquitetural, importante na convergência entre Computação Móvel e Computação em Nuvem. Ela representa o elemento intermediário em uma arquitetura de três camadas: dispositivo móvel – cloudlet – nuvem.

Uma *cloudlet* pode ser vista como como um “*datacenter in a box*”, cujo objetivo é trazer as facilidades de nuvens mais perto dos dispositivos móveis. Além disto, uma *cloudlet* pode ser vista com uma infra-estrutura local de nuvem, normalmente internalizada como uma máquina virtual. A Figura 19 ilustra um cenário possível de uso de *cloudlets*:



Figura 19. Cenário de uso de cloudlet.

Fonte: (Satyanarayanan et al., 2009)

Neste cenário, tem-se uma infra-estrutura local de nuvem (Coffee shop), que funciona como um mediador local para serviços mais complexos de nuvem (Yahoo!). Os diversos dispositivos móveis (óculos, celular, luva e *tablet*) acessam localmente facilidades como geo-localização, busca, serviços de data e hora, compartilhamento de fotos e vídeos via este mediador. Isto, permite, por exemplo utilizar esquemas de cache nas *cloudlets* para respostas mais rápidas ou mesmo utilizar a *cloudlet* como uma fachada comum para acessos a diversos recursos de nuvem mais complexos.

Assim, quando se decide pelo uso de *cloudlets*, três pontos precisam ser bem determinados:

- quais serviços estão disponíveis nos dispositivos
- quais serviços estão disponíveis nas *cloudlets*
- quais serviços estão disponíveis na nuvem

A decisão da localização dos serviços, geralmente, é determinada por requisitos de tempo. Latências de redes, muitas vezes, também impõem restrições de onde os serviços podem estar localizados.

3.4.2 Exemplo de Uso de Cloudlets em RA

Para exemplificar o uso de *cloudlets* em RA, será utilizado o cenário descrito por Verbelen *et al.* (2012) e mostrado na Figura 20:

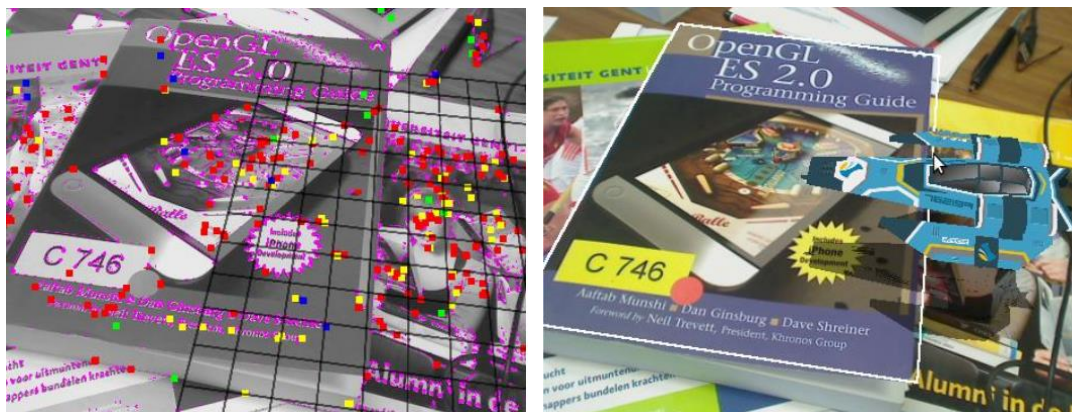


Figura 20. Serviços de tracking (esquerda) e fusão (direita)

Fonte: (Verbelen et al., 2012)

Neste cenário, há cinco serviços a serem disponibilizados:

- **VideoSource:** captura as imagens da câmera do dispositivo, gerando *frames* para serem analisados pelo serviço Tracker e fundidos com objetos 3D pelo serviço Renderer.
- **Renderer:** responsável por fundir objetos 3D com *frames*. Os objetos são posicionados de acordo com as posições estimadas de câmera geradas pelo serviço Tracker.

- **Tracker:** responsável por calcular estimativas de posições de câmera, com base em um mapa conhecido de pontos 3D gerado pelo serviço Mapper e nos *frames* gerados pelo serviço VideoSource.
- **Mapper:** de tempos em tempos, o serviço Tracker envia um *frame (keyframe)* para que o serviço Mapper, que gera um conjunto de pontos com coordenadas 3D baseado em *features* obtidas no frame.
- **Relocalizer:** quando não são encontradas features no frame, este serviço reposiciona a câmera para que o serviço Tracker possa continuar.
- **Object Recognizer:** nos keyframes recebidos pelo serviço Mapper, um Object Recognizer tenta localizar objetos conhecidos. Quando um objeto é reconhecido, sua localização 3D é informada ao Renderer para fusão.

Os serviços VideoSource e Renderer precisam ser executados no dispositivo. Já os serviços Tracker e Relocalizer têm restrições de tempo real (< 50 ms). Os serviços Mapper e Object Recognizer não têm requisitos estritos de tempo.

Assim, estes serviços podem ser agrupados conforme mostrado na Figura 21:

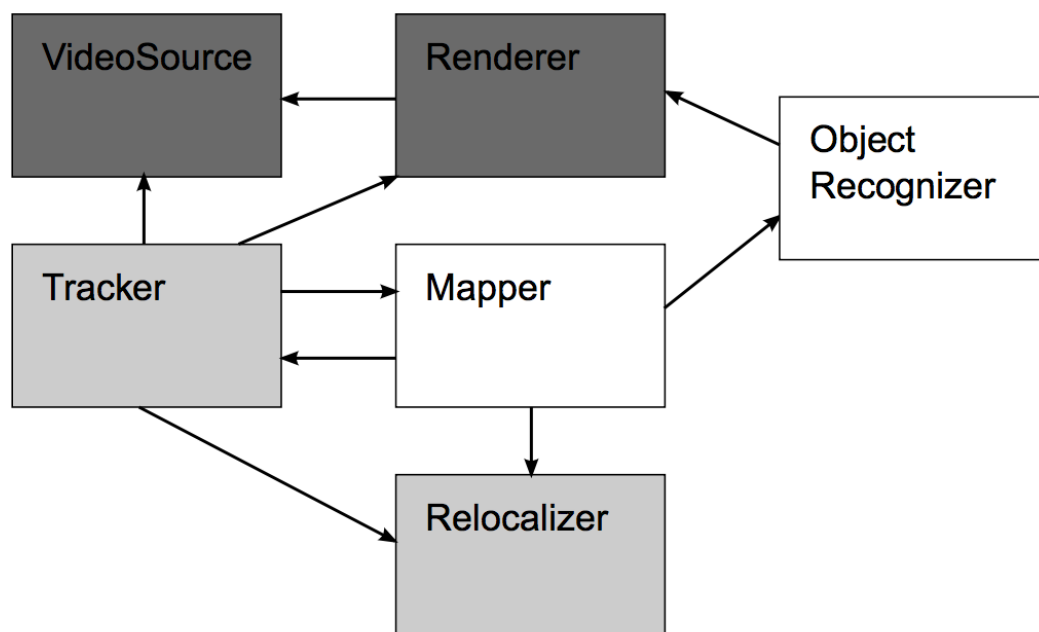


Figura 21. Agrupamento de serviços de RA
 Fonte: (Verbelen et al., 2012)

Os serviços VideoSource e Renderer serão executados no dispositivo. Os serviços Tracker e Relocalizer serão incorporado em uma *cloudlet* e, Mapper e Object Recognizer, em uma infra-estrutura de nuvem.

O esquema de implantação destes três grupos é mostrado na Figura 22:

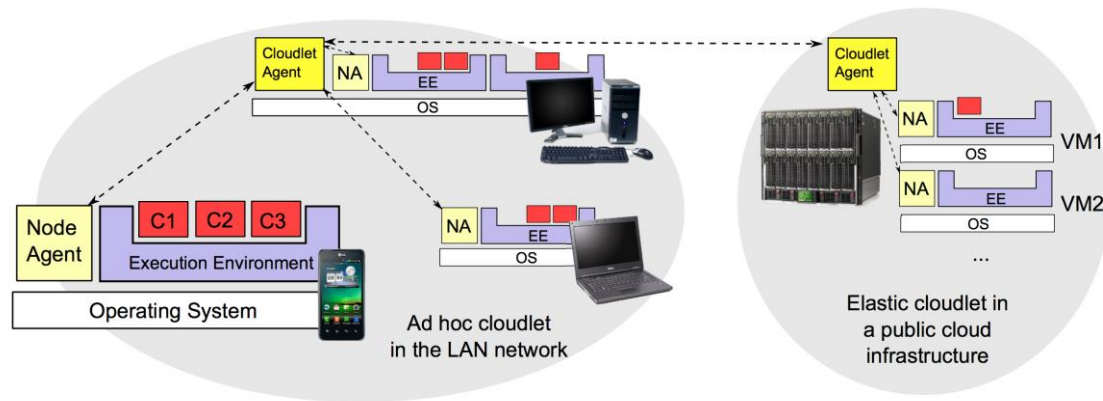


Figura 22. Implantação de serviços de RA nos dispositivos, cloudlets e nuvem

Fonte: (Verbelen et al., 2012)

Em cada ambiente (dispositivo, *cloudlet* e cloud) há um agente de comunicação, responsável pela integração entre as diversas partes. Para que a comunicação seja comum, é importante a utilização de envelopes de comunicação baseados em XML. Observe, também, que na nuvem pode-se utilizar a organização dos serviços como máquinas-virtuais (VM) e o esquema de acesso também pode ser organizado como uma *cloudlet*. Neste contexto, *cloudlets* da rede local são chamadas *cloudlets* ad hoc e, aquelas localizadas na nuvem, de *cloudlets* elásticas.

3.5 Comentários Finais

Mesmo com o aumento do poder computacional e de armazenamento dos dispositivos móveis nos últimos anos, alguns processos de RA ainda precisam estar localizados em estações com capacidade computacional e armazenamento mais potentes. Dentre deste contexto, este pequeno texto introdutório teve o objetivo de apresentar a tecnologia de *cloudlets* como uma arquitetura viável para distribuição de processamento e armazenamento para aplicações de RA. Os autores esperam que ela possa ser um ponto inicial de suporte para alunos, professores e pesquisadores que se interessem por aplicações de RA distribuídas, com suporte a nuvem.

Referências

- Azuma, R.T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*. 6 (4), Ago 1997.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. *Above the clouds: A Berkeley view of cloud computing*. Technical report, 2009, EECS Department, University of California, Berkeley.

Buyya, R., Vecchiola, C., Selvi, S.T. **Mastering Cloud Computing: Foundations and Applications Programming**. New York: Morgan Kaufmann, 2013.

Erl, T., Puttini, R., Mahmood, Z. **Cloud Computing: Concepts, Technology & Architecture**. New York: Prentice Hall, 2013.

Gatech. **UART – Unity AR Toolkit**. URL: <https://research.cc.gatech.edu/uart/>. Acessado em Março/2014.

Gart. **Gart – Geo AR Toolkit**. URL: <http://gart.codeplex.com/>. Acessado em Março/2014.

Gartner. **Gartner IT Glossary - Cloud Computing**. URL: <http://www.gartner.com/it-glossary/cloud-computing/>. Acessado em Março/2014.

Google Code. **AndAR – Android Augmented Reality**. URL: <https://code.google.com/p/andar/>. Acessado em Março/2014.

Hwang, K., Dongarra, J., Fox, G. C. **Distributed and Cloud Computing: From Parallel Processing to the Internet of Things**. New York: Morgan Kaufmann, 2011.

Linthicum, D. **Defining the Cloud Computing Framework**. URL: <http://cloudcomputing.sys-con.com/node/811519>. Acessado em Março/2014.

Lisboa, M. **Computação na Nuvem**. URL: <http://marcosplisboa.wordpress.com/2011/05/03/61/>. Acessado em Março/2014.

NIST. **The NIST Definition of Cloud Computing**. NIST, 2011. URL: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Acessado em Março/2014.

Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N. The Case for VM-based Cloudlets in Mobile Computing. **IEEE Pervasive Computing**, 8(4), Outubro, 2009, pp.2—11.

Vaquero, L.M., Roderio-Merino, L., Caceres, J., Lindner, M. A break in the clouds: towards a cloud definition. **SIGCOMM Comput. Commun.** 39 (1), Dez 2008, pp. 50-55.

Verbelen, T., Simoens, P., Turck, F., Dhoedt, B. Cloudlets: Bringing the Cloud to the Mobile User. **Proceedings of III ACM Workshop on Mobile Cloud Computing and Services(MCS'2012)**, LowWoodBay, Reino Unido, 2012, pp. 29—36.

Youseff, L.; Butrico, M.; Silva, D. Toward a Unified Ontology of Cloud Computing. **Grid Computing Environments Workshop GCE'08**, 1(10), Nov. 2008, pp.12-16.

Zhang, Q., Cheng, L., Boutaba, R. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**. 1 (1), Springer-Verlag, 2010, pp.7-18.