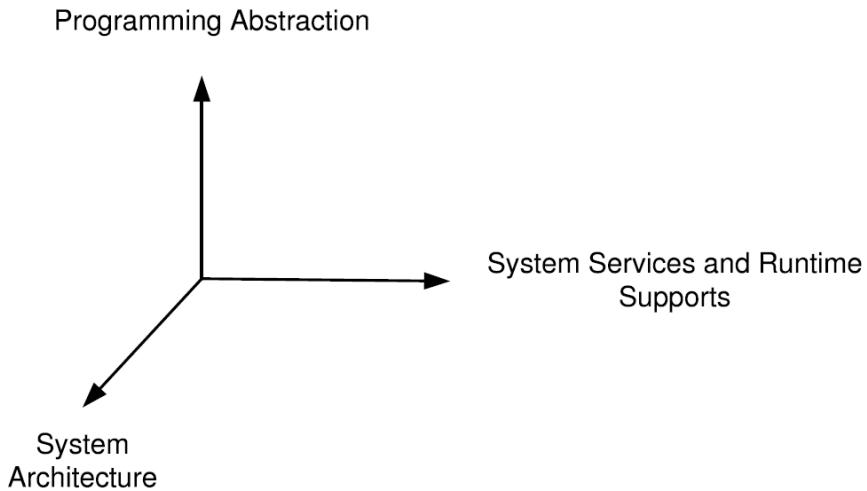




Sensibilidade ao Contexto Definições

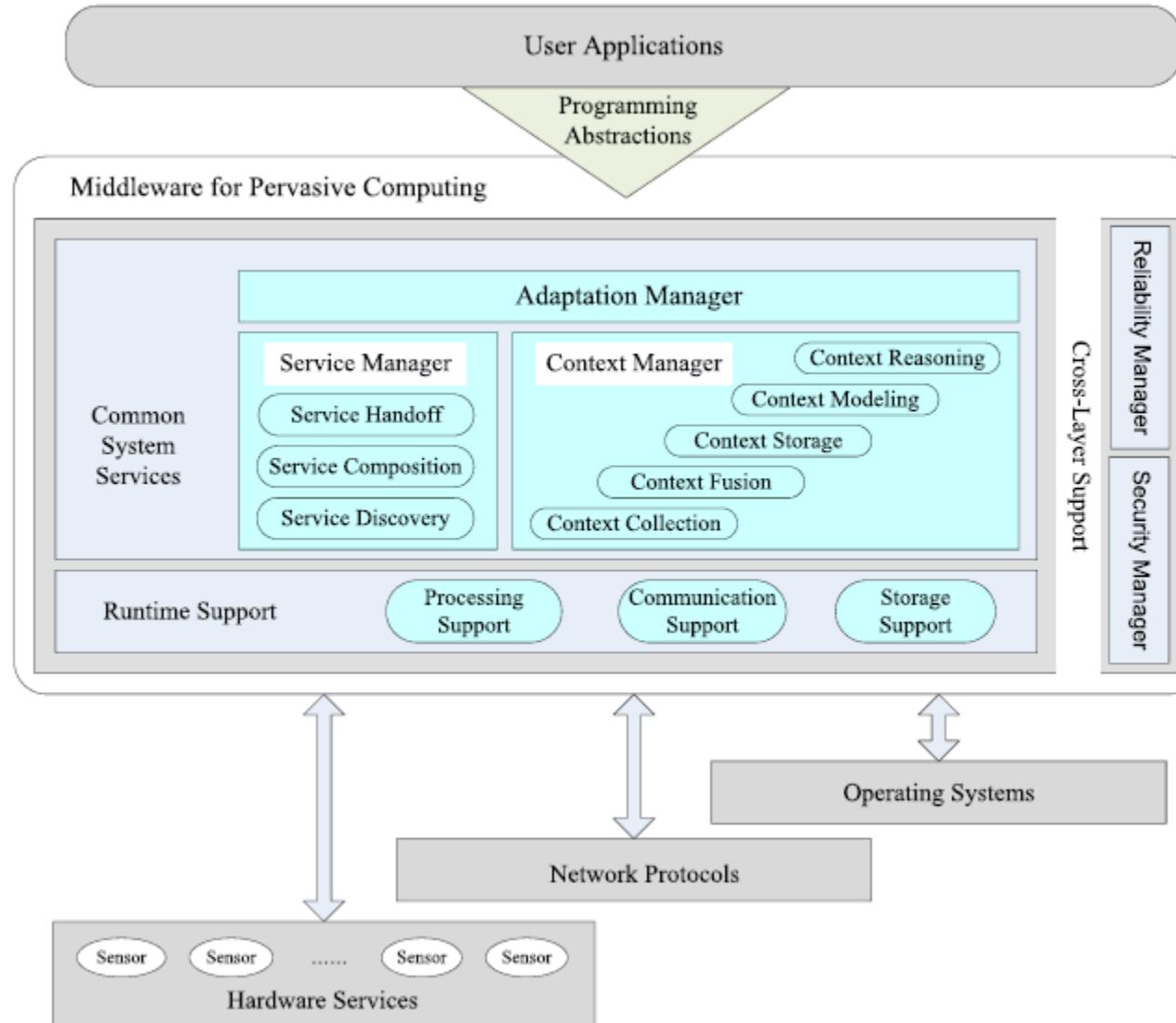
Professores Windson Viana e Fernando Trinta
Disciplina de Computação Móvel e Ubíqua
Curso de Sistemas e Mídias Digitais

Dimensões do Design de um Middleware para sistemas ubíquos/pervasivos



Vaskar Raychoudhury, Jiannong Cao, Mohan Kumar, Daqiang Zhang,
Middleware for pervasive computing: A survey,
Pervasive and Mobile Computing, Volume 9, Issue 2, April 2013, Pages 177-200

Arquitetura de Referência



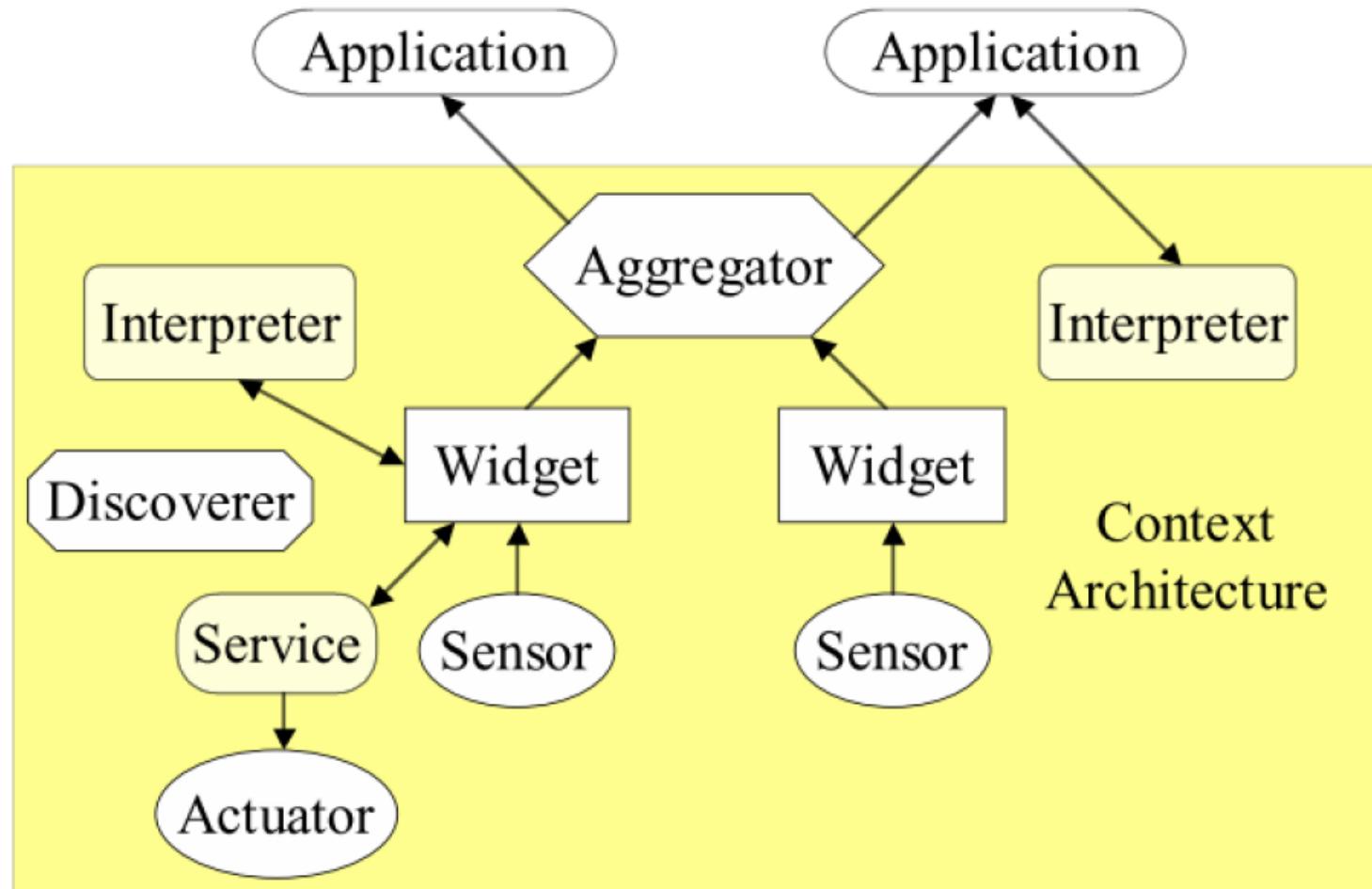
De onde isso começou?

Context Toolkit (Dey, 1999)

- Proposição de um framework orientado a objetos
- Separação em camadas de conceitos recorrentes
- Simples demais, porém serviu como um marco inicial

- <http://contexttoolkit.sourceforge.net/>

Context Toolkit (Dey, 1999)



Context Toolkit

Exercício

- Quais os papéis de cada tipo de componente?
- Diferencie um Agregador de um Interpretador
- Que tipo de informação a aplicação pode receber?

CASS (FAHY; CLARKE, 2004)

- Arquitetura Cliente-Servidor
 - Definição de papéis distintos para dispositivos móveis, sensores e um servidor
 - Baseado no poder e nas restrições de processamento
- Sensores fornecem dados básicos e Dispositivos móveis são consumidores das informações produzidas no servidor
- Ideia de um banco de dados para armazenamento de histórico de dados contextuais

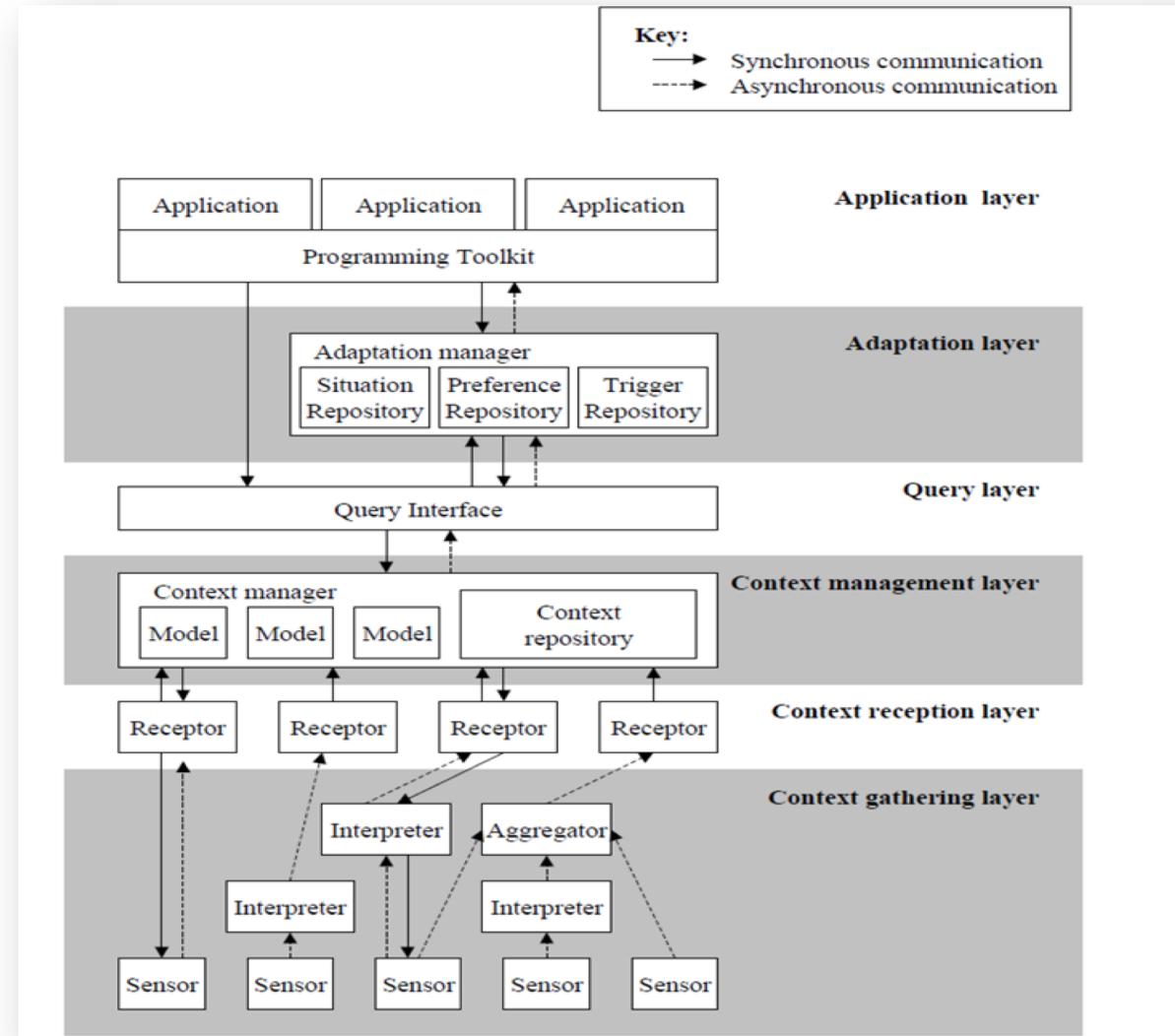
Context Management Toolkit

[Henricksen et al., 2004]

- Desenvolvimento de aplicações sensíveis ao contexto
 - Organização em camadas
 - Captura, interpretação e inferência de contexto
 - Separação entre adaptação e gerenciamento de contexto
- Aplicações acessam ou se subscrevem no middleware para obtenção de informações contextuais
 - Uso de um banco de dados “orientado a objetos”

Context Management Toolkit

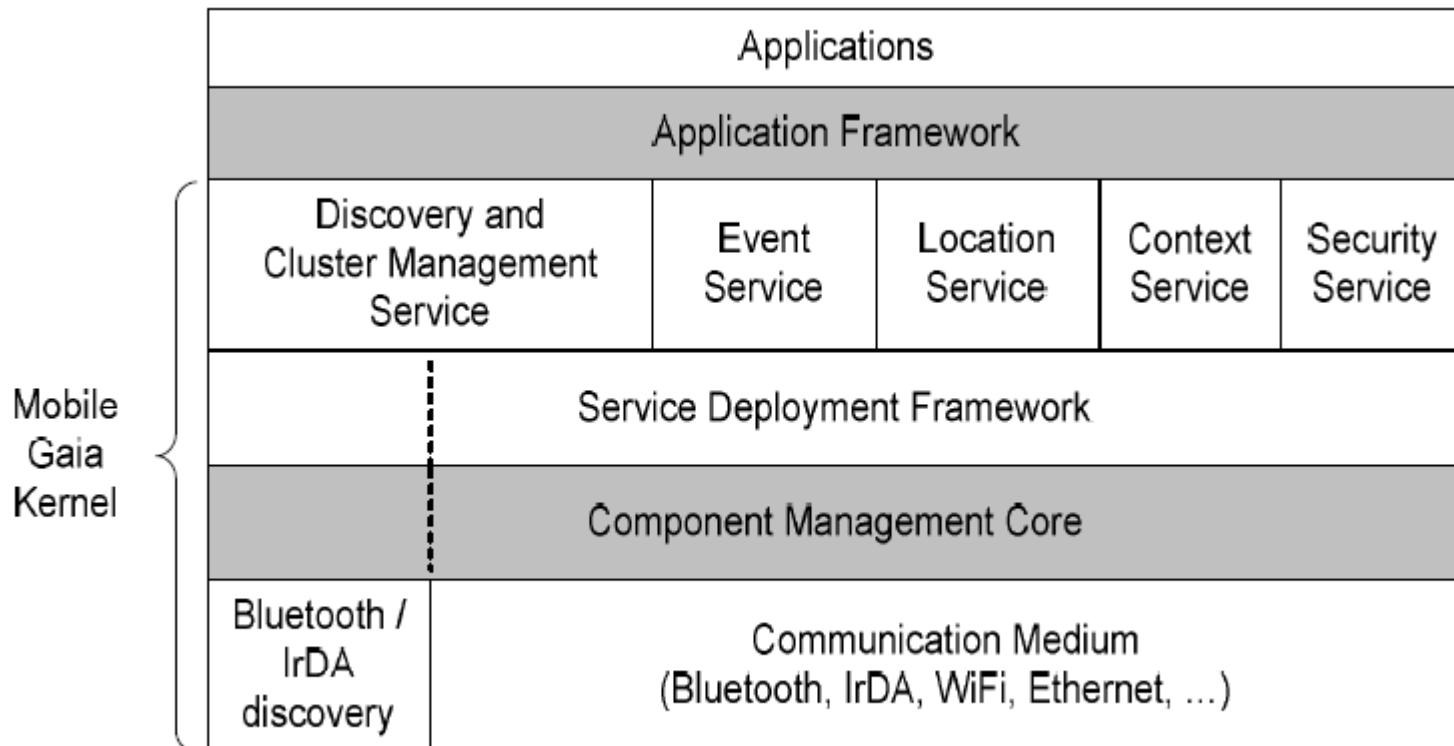
[Henricksen *et al.*, 2004]



Mobile Gaia (Chetan et al., 2005)

- Middleware orientado a serviço
 - Integração e comunicação entre vários dispositivos
 - A noção de cluster de dispositivos
- Modelo de programação
 - Divisão das aplicações em componentes
 - API para programação
 - Descoberta de Serviço e formação de clusters

Mobile Gaia (Chetan et al., 2005)



Mas afinal o que é um
middleware?

Middleware

- Middleware mascara heterogeneidade de sistemas distribuídos
 - Java VM: SO
 - CORBA: Linguagem
 - OSGi
 - Web Services REST
- Trata de concorrência, portabilidade, interoperabilidade
 - API única, protocolo comum, conjunto de serviços
- Middlewares tradicionais facilitam a vida do desenvolvedor pela abstração de transparência.

Integração usando Middleware de comunicação

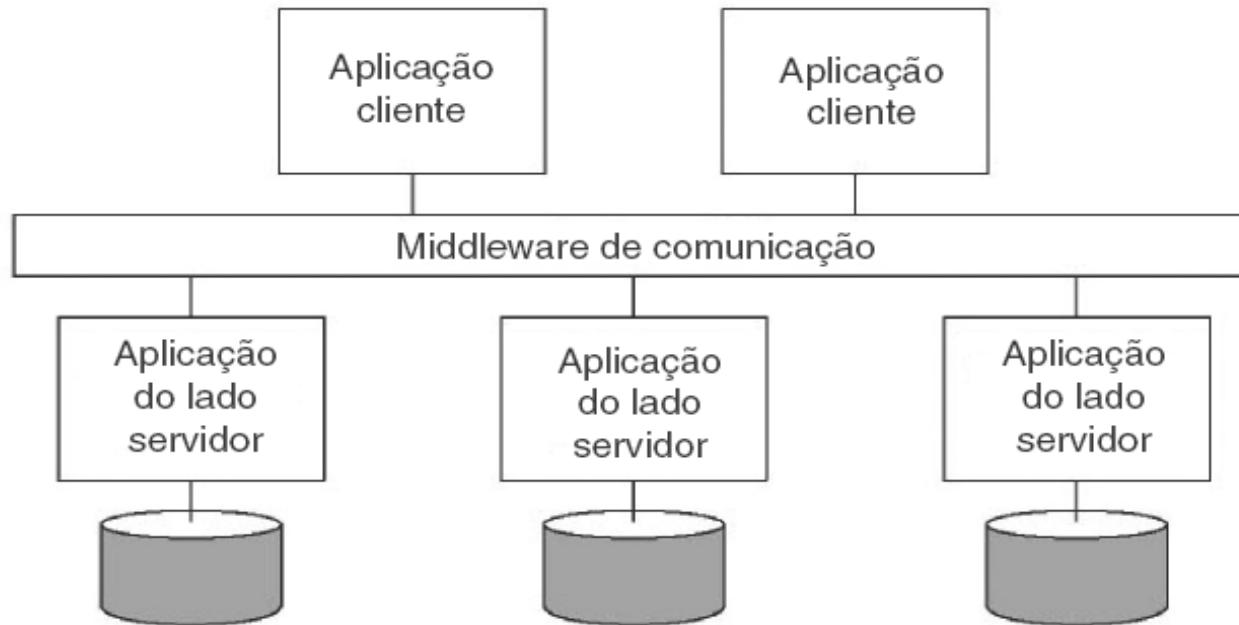
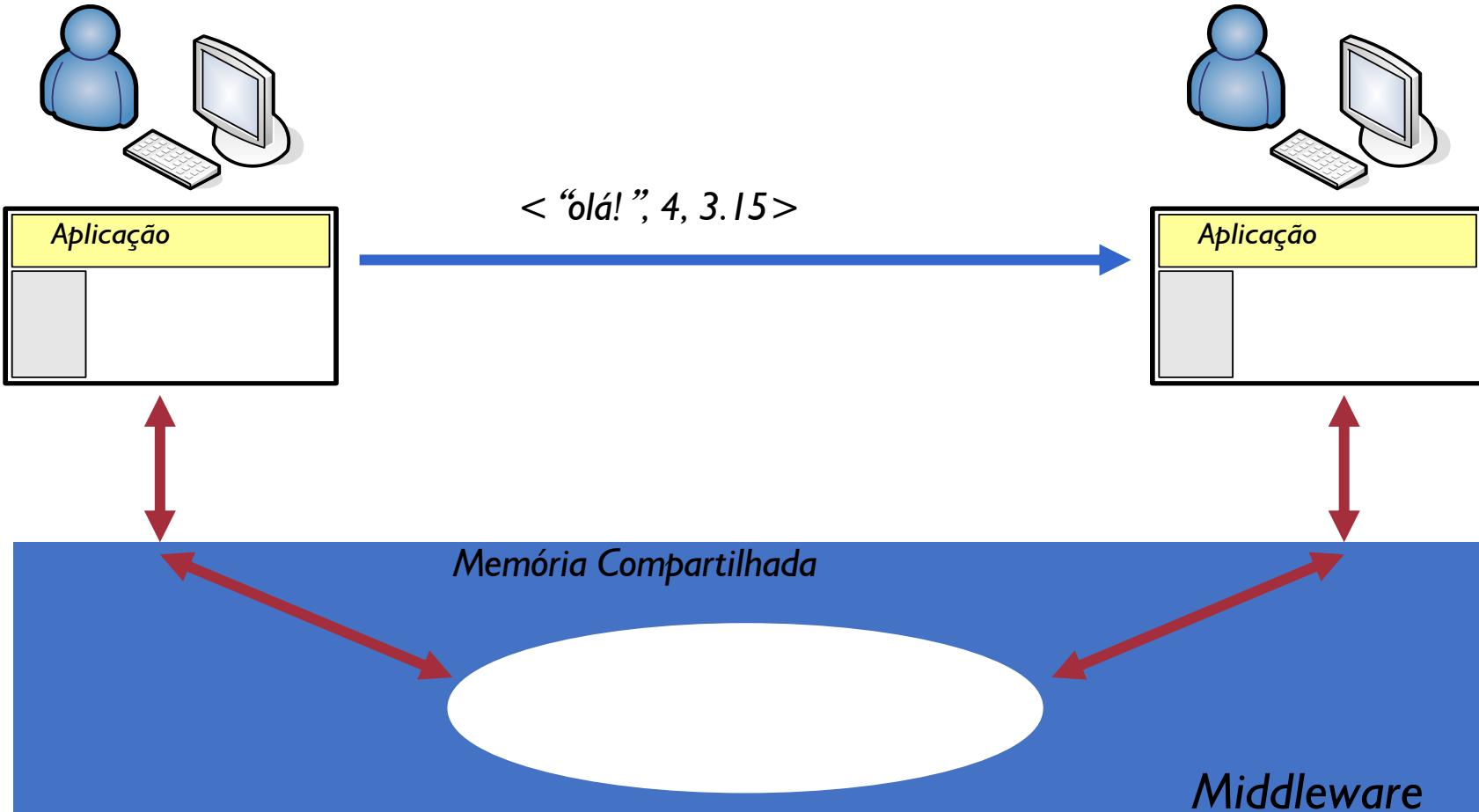


Figura 1.8 Middleware como facilitador de comunicação em integração de aplicações empresariais.

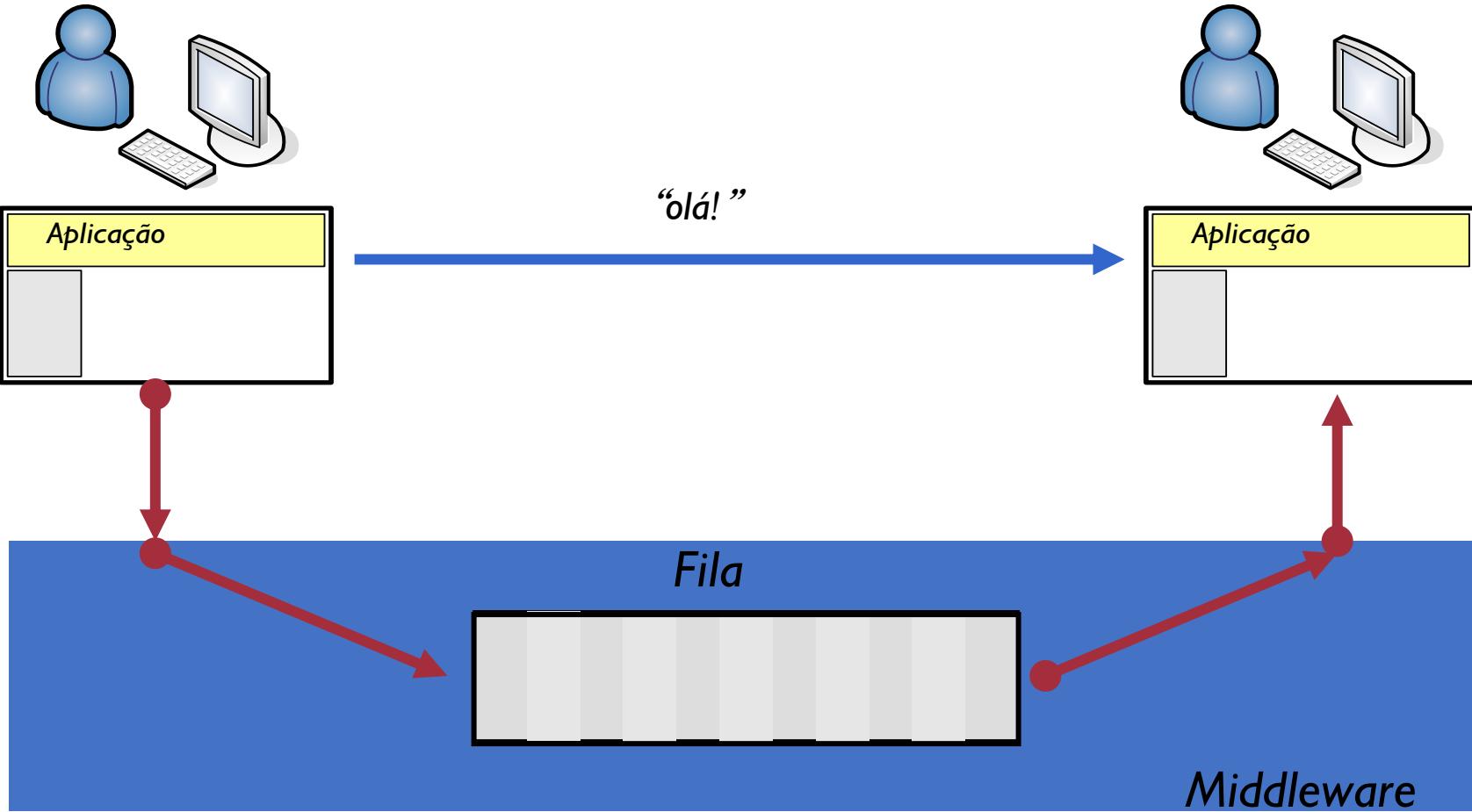
Modelos de Middleware Tradicionais

- Middleware orientados a transação
- Middleware com memória compartilhada
- Middleware orientados a mensagem
- Middleware orientados a RPC
- Middleware orientados a Objetos
- Middleware Adaptativo
- Middleware Multimídia

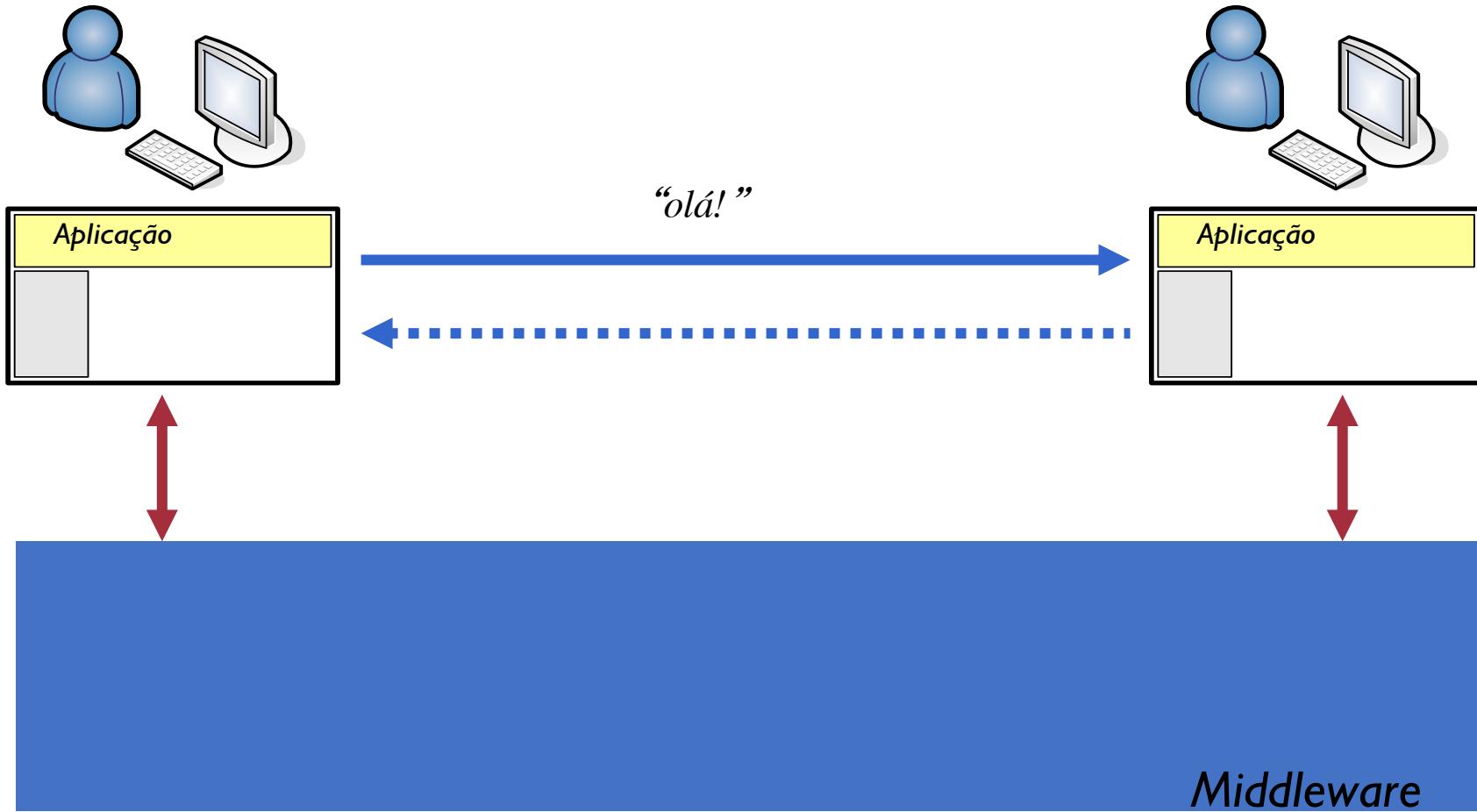
Memória Compartilhada



Orientado a Mensagem



RPC/Objetos



Oito requisitos para middleware ubíquos

[Ibrahim et al, 2010]

- Interoperabilidade
- Descoberta de Dispositivos e Serviços
- Transparência de Localização
- Adaptabilidade
- Sensibilidade ao Contexto
- Escalabilidade
- Segurança
- Gerenciamento autônomo

Middleware for Context-Aware Applications

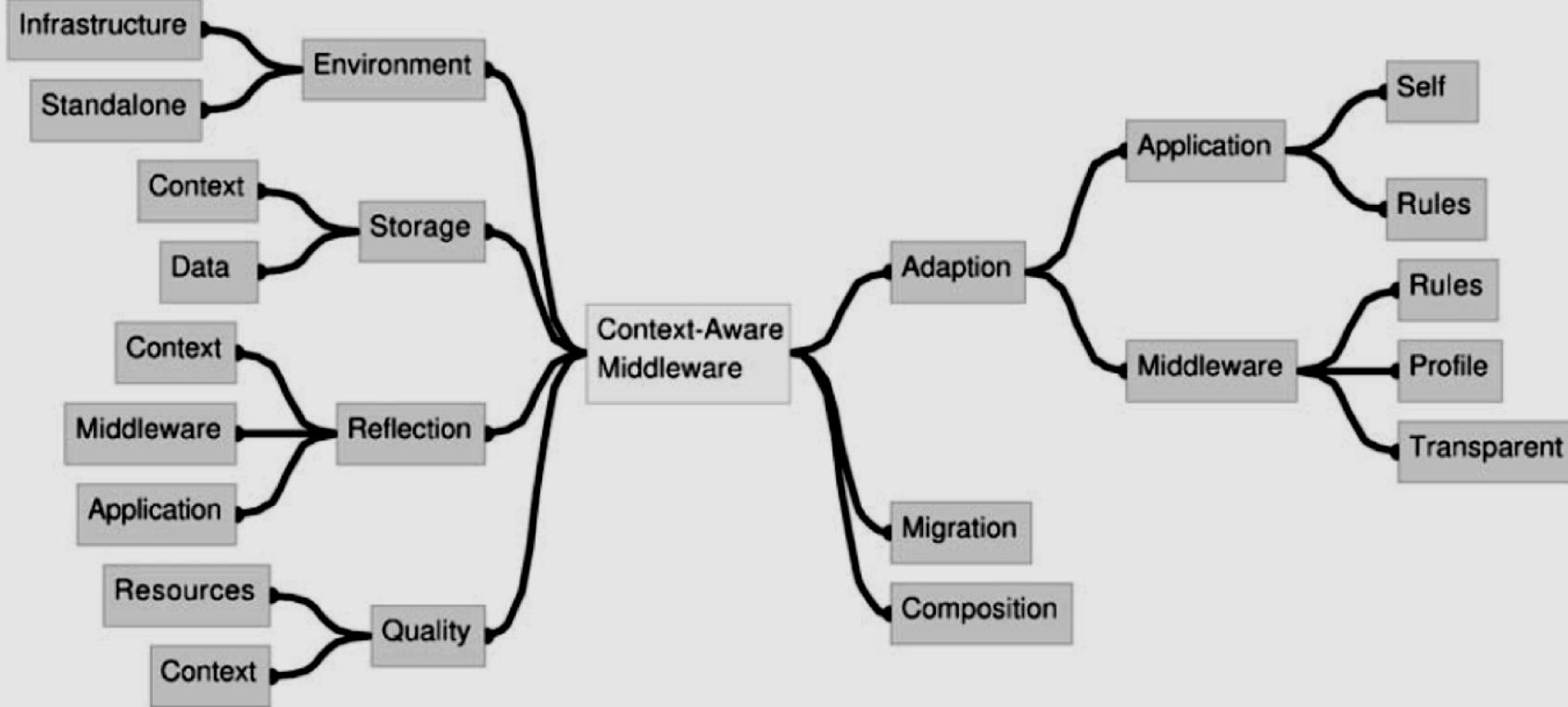


Figure 1. Taxonomy of Context-Aware Middleware showing major categories, sub categories, and sub-sub categories

Em visão em camadas

Aplicação

Contextual

Lógica

Física

- Cada camada fornece um nível de abstração
 - De acordo com as abstrações anteriores
 - Modelo comum
- Interação entre camadas através de uma interface bem definida e protocolos padronizados.
 - Mudanças nas camadas são possíveis, uma vez que as interfaces são mantidas

Em visão em camadas



- Sensores captam sinais puros
 - GPS (localização)
 - Sensores de pressão, temperatura, luminosidade
 - Câmeras ou microfones

Em visão em camadas



- Modelagem em software da camada física
 - Codificação de sinais para dados e eventos primitivos
 - Interpretação básica de valores. Ex: (TempSensor = 100 então Tempo = Quente)
- Aspectos a serem modelados:
 - Precisão
 - Confiabilidade
 - Granularidade

Em visão em camadas



- Transformação de dados dos sensores em dados de contexto
- Fusão/Interpretação de dados dos sensores
 - Combinação de dados de diferentes tipos
 - Leitura de vários dados de sensores ao longo do tempo
 - Inferir Probabilidade a partir de uma Função F (Precisão, granularidade, confiabilidade)
- Definição de contextos complexos
 - Dirigindo

Em visão em camadas



- Definição do contexto apropriado/relevante para a aplicação
 - Combinação de contextos de diferentes naturezas
 - Definição de ações a serem tomadas em quanto de um contexto

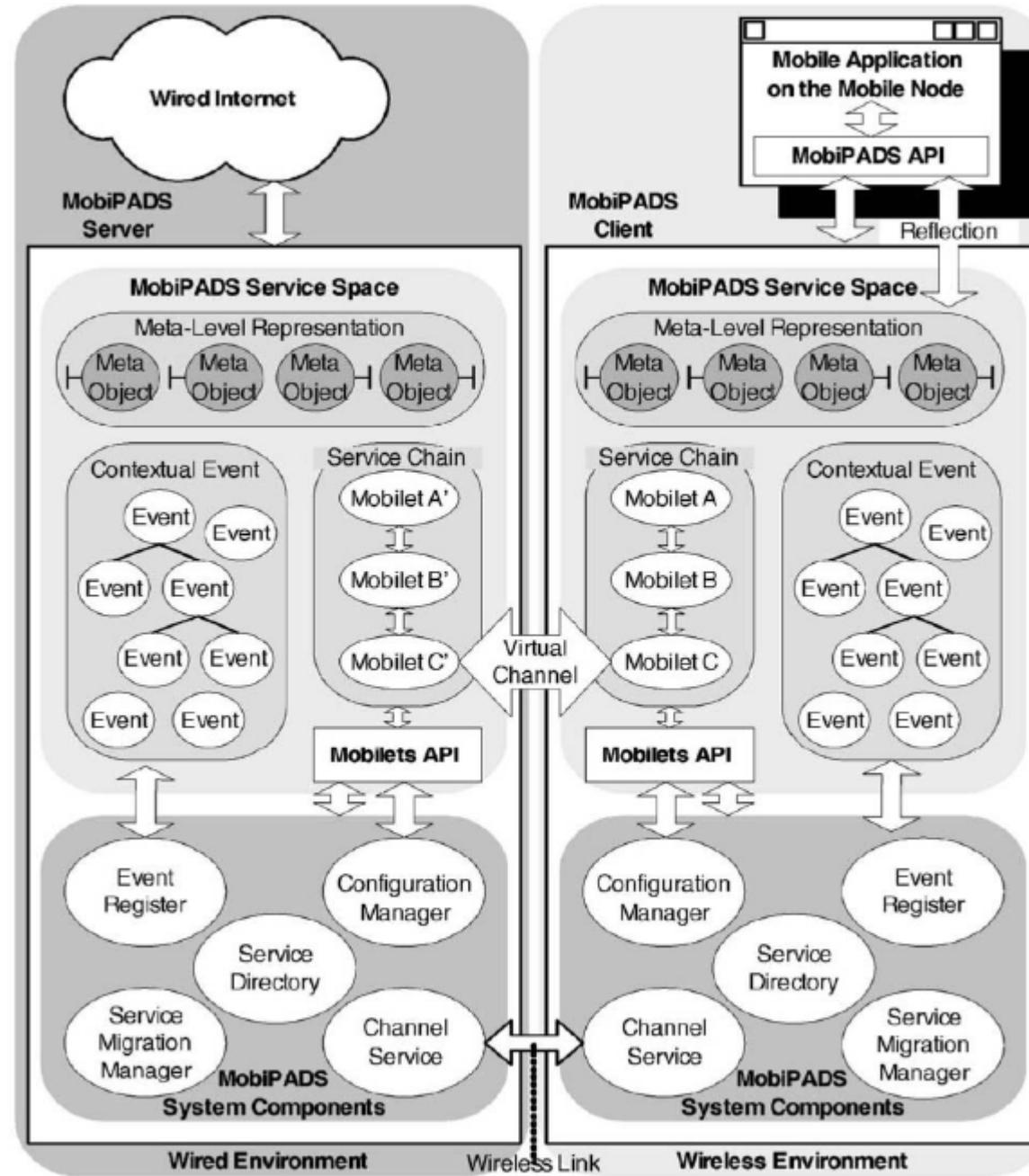
Problemas

- Privacidade
 - Usuário precisa disponibilizar sua informação de contexto
- Precisão e atualidade
 - Se o processamento de contexto é muito demorado ou impreciso, então é de pouco valor para a maioria das aplicações
- Autenticidade
 - Como garantir que a informação de contexto de um usuário é autentica (não foi manipulada por ele ou outros) ?
- Eficiência Energética

Exemplos

MobiPads (Chan et al., 2005)

- Middleware reflexivo para J2SE
 - Modelo Arquitetural Cliente-Servidor
- Aplicações descritas como cadeias de serviços em um documento XML
 - Mobilets
- Serviço de contexto baseado em eventos
- Reconfiguração dinâmica provida pelo middleware



SOCAM

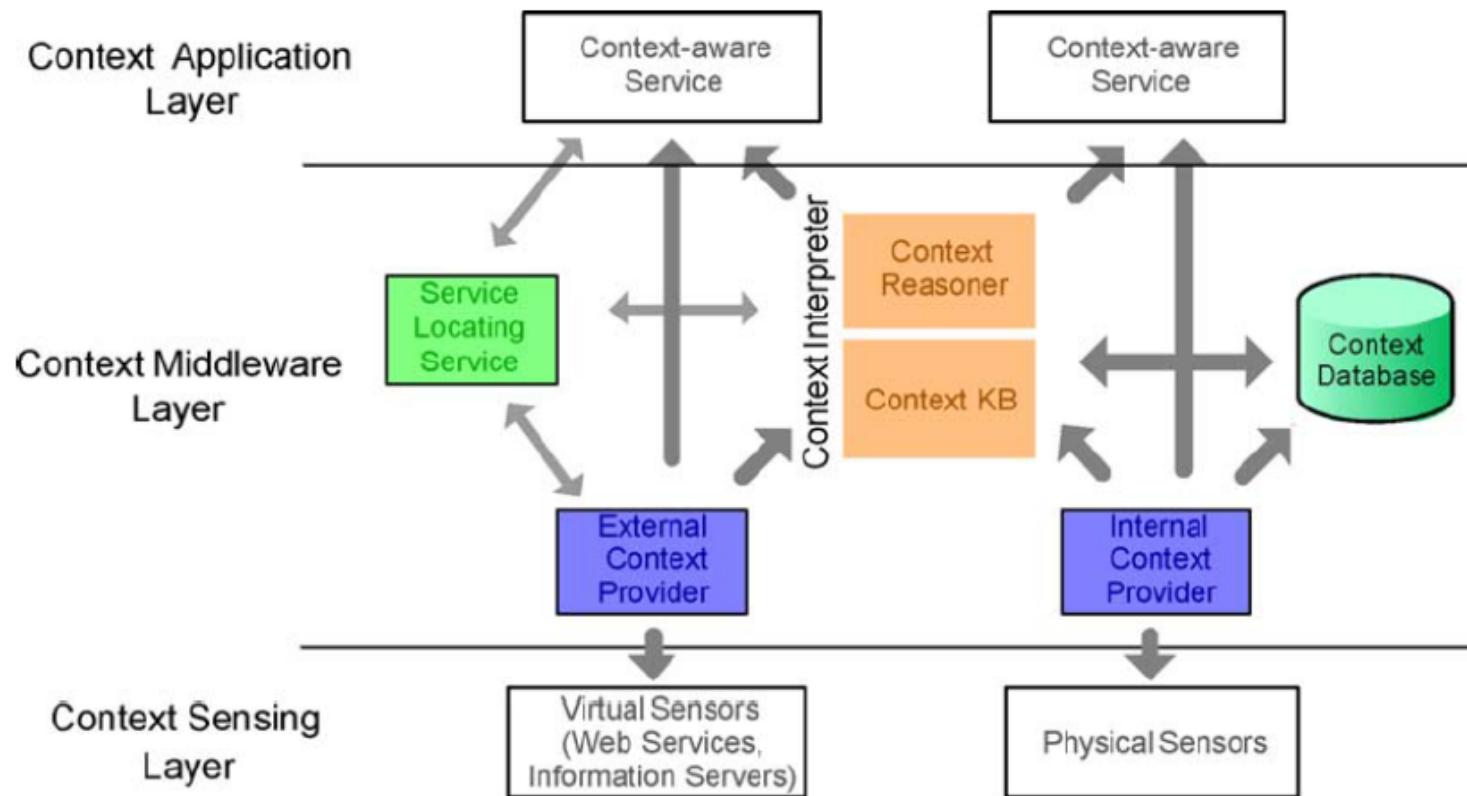
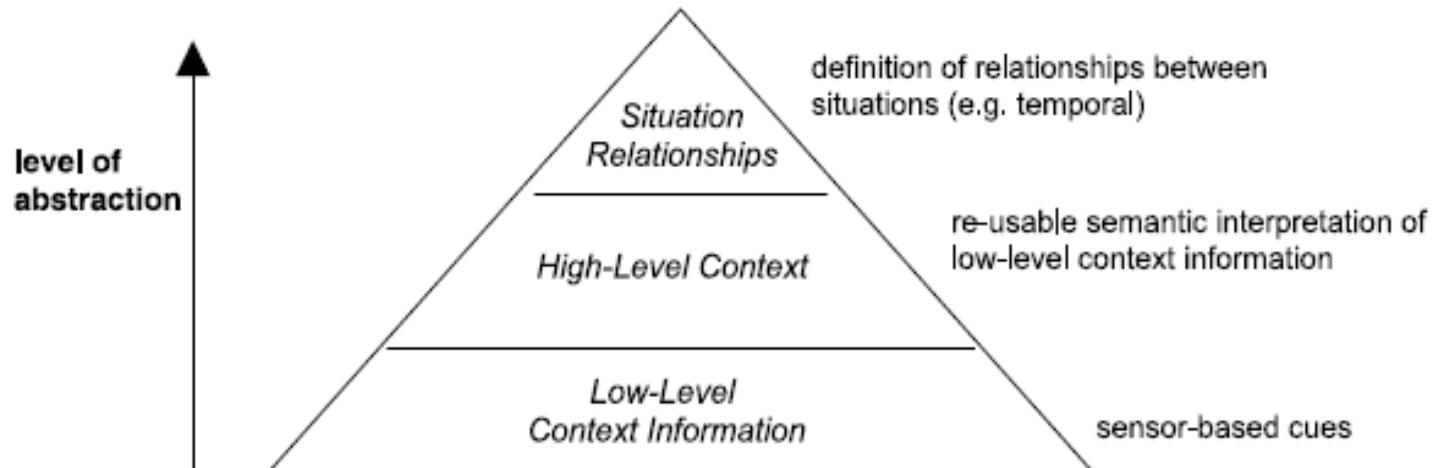


Figura 3.3: Arquitetura do SOCAM (GU et al., 2005).

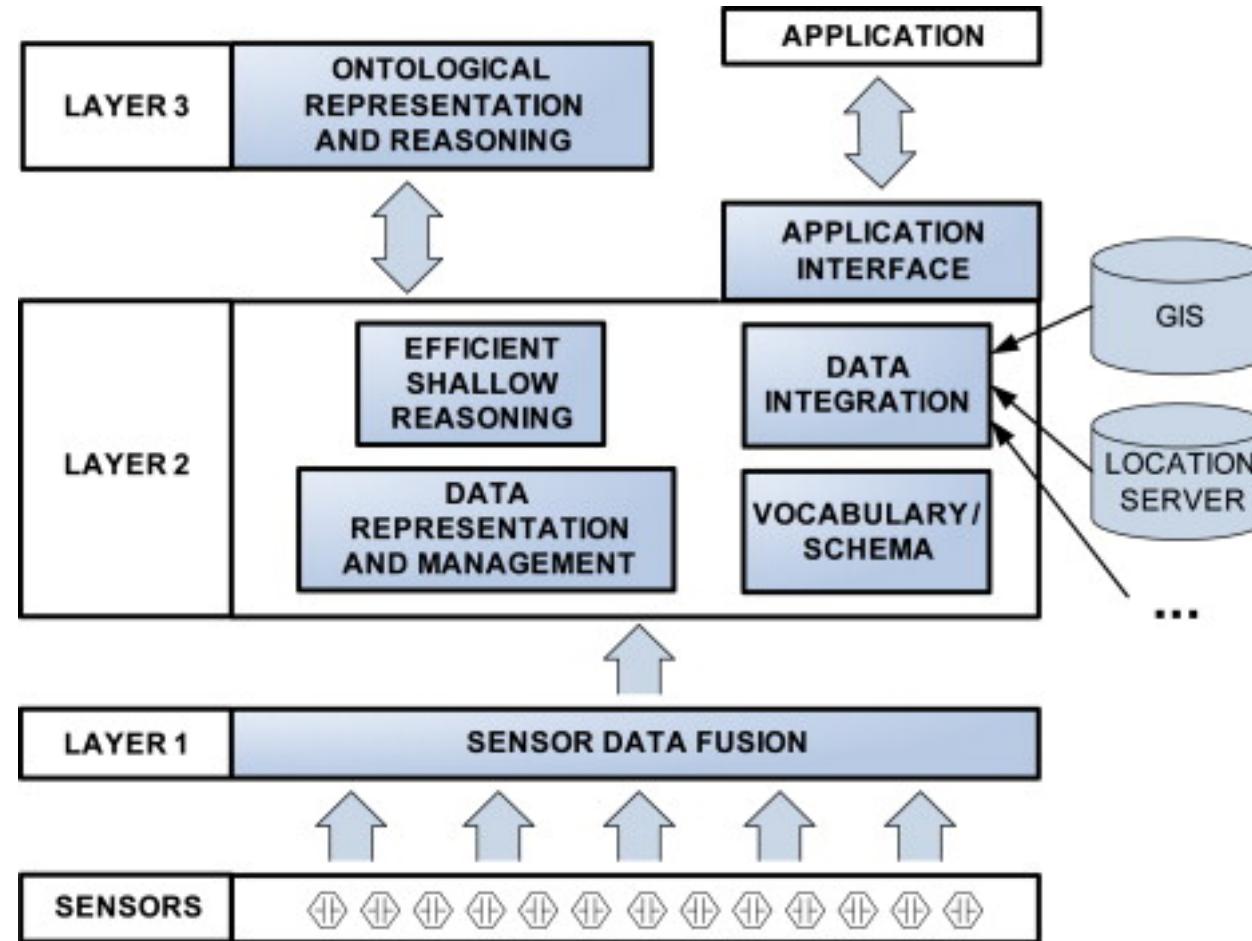
Multilayer Framework

[Bettini et al, 2010]

- Evolução do trabalho de Henricksen
 - Representação de contexto utilizando tecnologias da Web Semântica
 - Ontologias, Regras SWRL e modelos para tratar incerteza
 - Focado em prover informação contextual de alto nível semântico



Multilayer Framework [Bettini et al, 2010]



NTT Docomo - 2012

ECA-DL

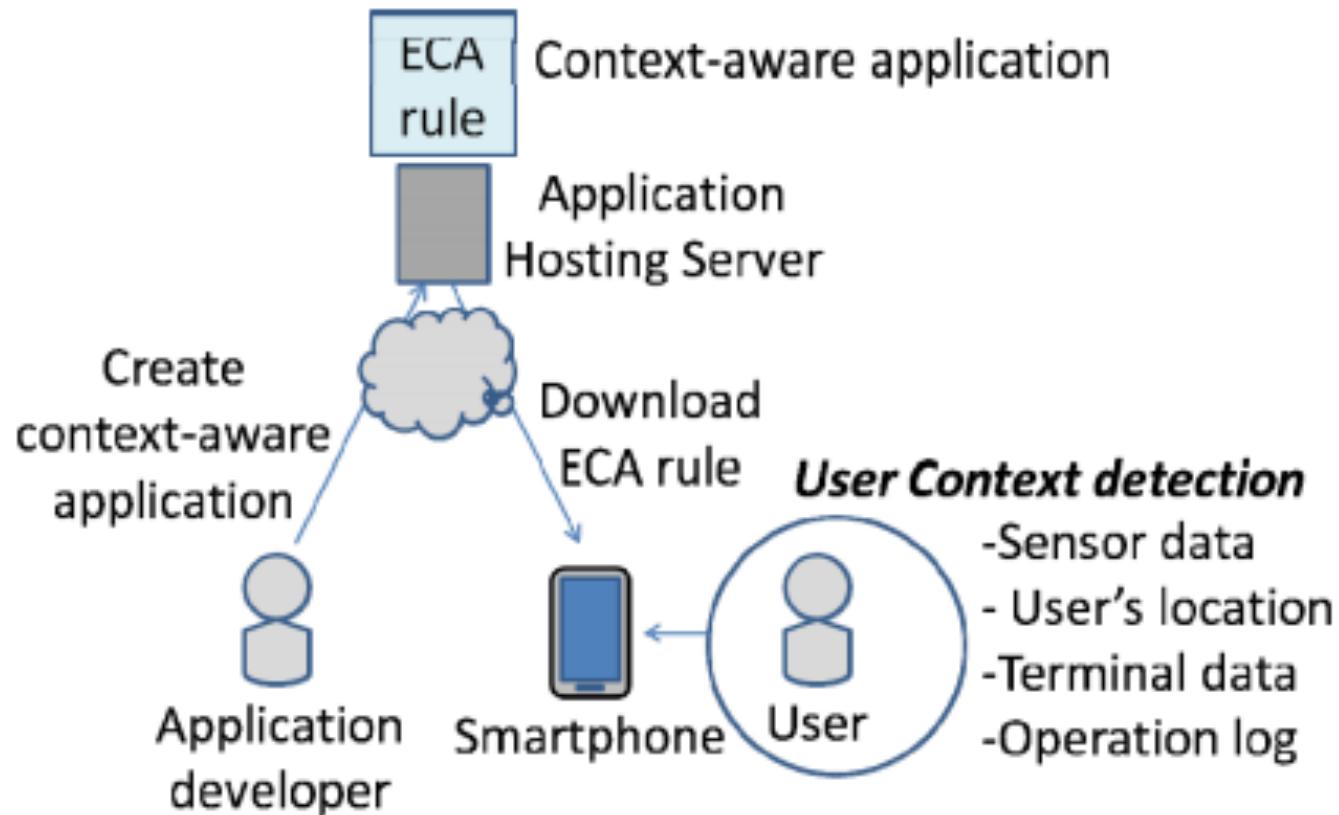


Figure 1: User context detection by ECA rule

NTT Docomo - 2012

ECA-DL

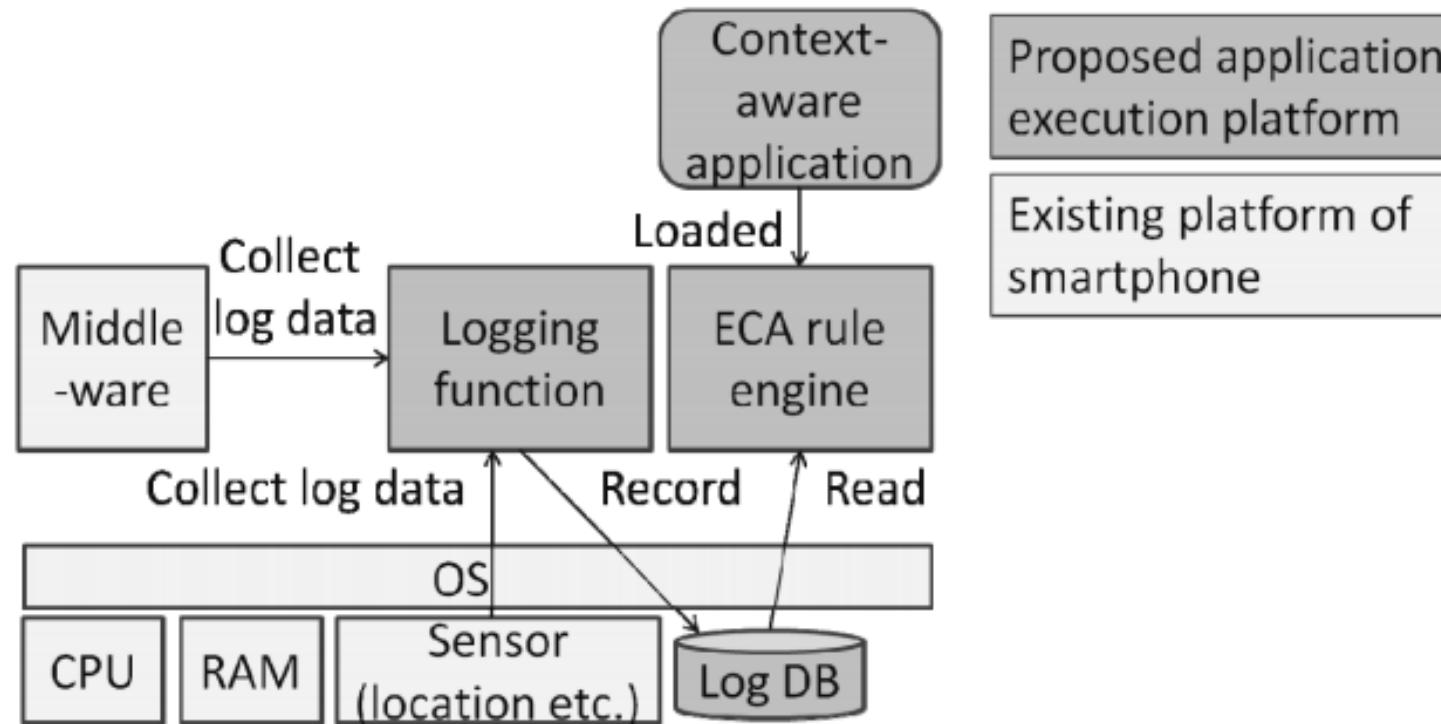


Figure 2: System configuration of the platform

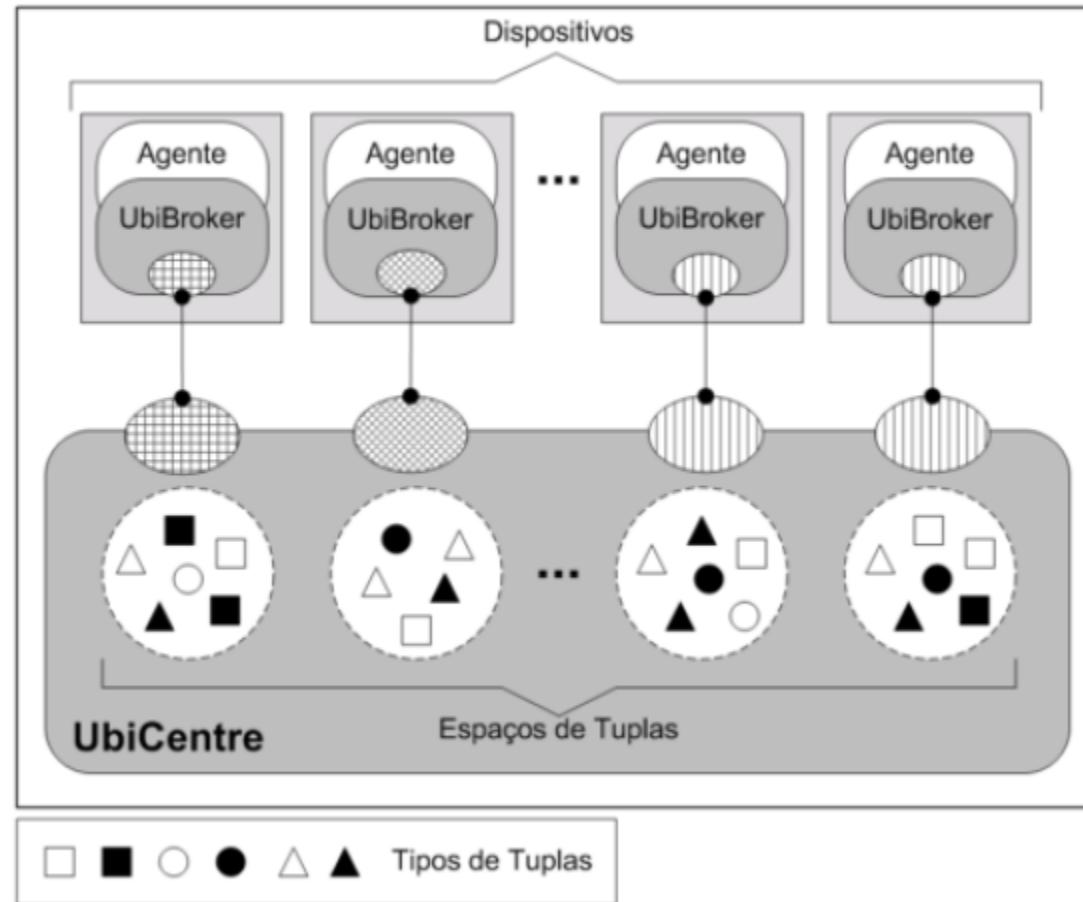
SySSU (Great-2010)

- Infraestrutura para implementação de coordenação desacoplada entre sistemas ubíquos
 - Desenvolvido no Great como dissertação do Fabrício Lima
- Todo o funcionamento dessa infraestrutura é baseado em tuplas.
 - Uma tupla é uma composição de um conjunto de campos chave/valor.
- Existem duas formas para que aplicações possam utilizar informação publicadas no SysSU.
 - A primeira é realizar uma consulta direta. Isso exige que as informações tenham sido disponibilizadas nele antes da consulta ser realizada.
 - A Segunda forma é a baseada em eventos (publicação/subscrição). Ela é utilizada quando a informação ainda não está disponível, mas uma aplicação deseja ser notificada quando estiver.
- As duas formas de acesso a informações fazem uso de templates e filtros para selecionar as tuplas desejadas.

SySSU

- No SysSU uma tupla é formada por um conjunto de campos do tipo chave/valor
 - `{(name, "roberto"), (location, "casa")}`
- O UbiCentre (uma das implementações) oferece importantes funcionalidades sobre o conjunto de tuplas que formam seus espaços de tuplas
 - mecanismo de associação (Matching)
 - mecanismo de eventos (Reactions)

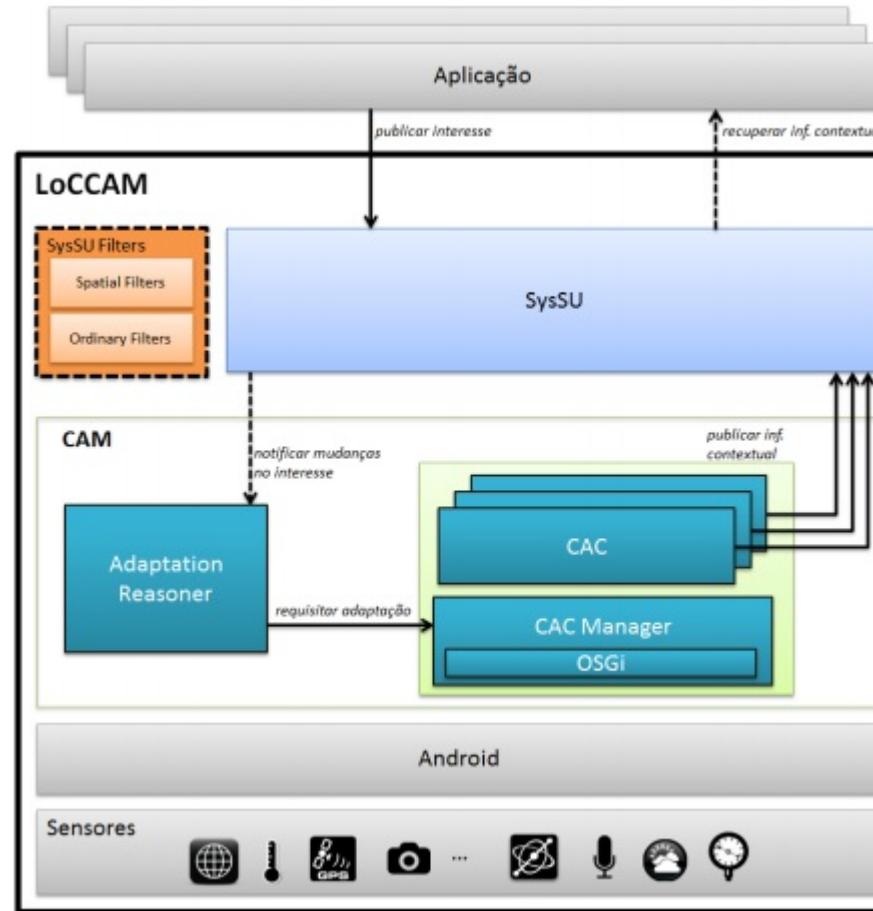
Ubicentre-UbiBrokers (SySSU)



LoCCAM (Great - 2014)

- Hipótese: dispositivo móvel como centro de aquisição e uso de contexto
 - Middleware para dispositivos móveis
 - Focado em prover abstrações de aquisição e detecção de contexto
 - Adaptação da camada de aquisição automática
- Modelo de programação baseado em espaço de tuplas

LoCCAM



Filter Tuple

```
public boolean filter(Tuple tuple) {
    for (int i = 0; i < tuple.size(); i++) {
        if(tuple.getField(i).getName().equals("Values")) {
            List<String> stringValues = (List<String>)tuple.getField(i).getValue();
            List<Double> values = new ArrayList<Double>();

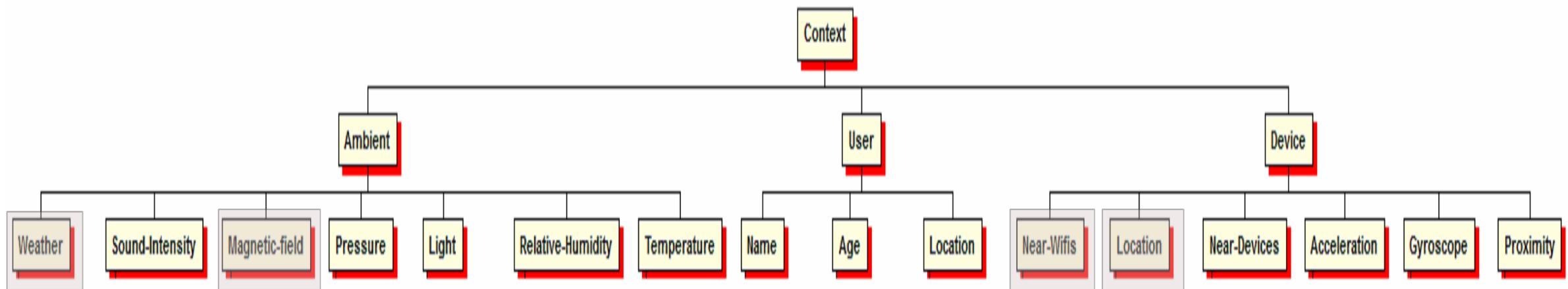
            for (String value : stringValues)
                values.add(Double.parseDouble(value));

            userLocation = new Location("");
            userLocation.setLatitude(values.get(0));
            userLocation.setLongitude(values.get(1));
        }
    }

    return location.distanceTo(userLocation) <= distance;
}
```



CACs Suportados

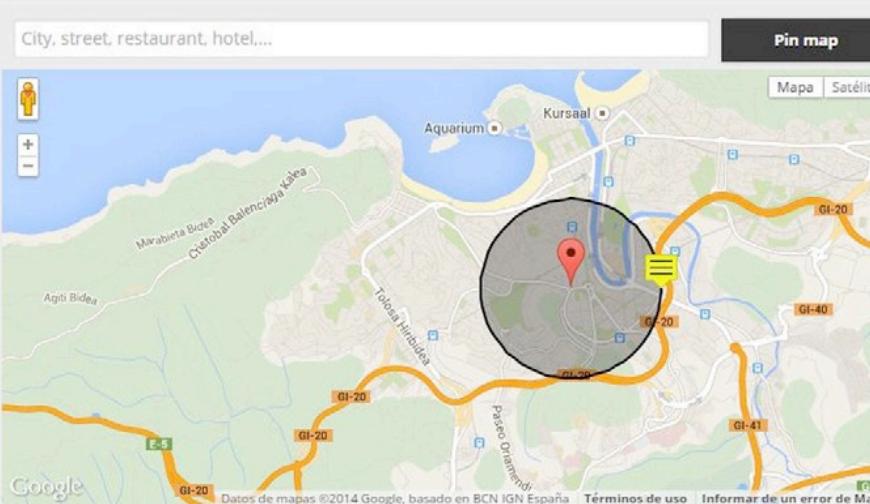


Cloud-based middleware (2015)

Areas

Create

City center ➤



City, street, restaurant, hotel,...

Pin map

Mapa **Satélite**

Name of area **Type of area**

City center **outdoors**

Every time someone arrives at this area

date between dd/mm/aa: ▾ and dd/mm/aa:

hour between 7 24

Save **Delete**

Information

es fr en

Title
Welcome!

Normal

The city center is a great place for shopping

Icons

!	★	ⓘ	🍴	☕
👍	📍	📷	🛎	🏋️
👤	🅿️			

Comparação - Yürür et al. 2016

TABLE II
NOTABLE CONTEXT-AWARE MIDDLEWARE STUDIES

Reference	Middleware	Architecture	Context Management	Context Source	Context Level	Reflection	Adaptation	Security and Privacy
[45]	Context Toolkit	Centralized	Key-value	Any type	High	-	Application based	✓
[46]	Aura	Distributed	Mark-up	Any type	High	✓	Middleware based	-
[47]	CARISMA	Distributed	Mark-up	Mobile	High	✓	Middleware based	-
[48]	Gaia	Distributed	Logic and Ontology	Any type	High	-	Application based	✓
[49]	SOCAM	Distributed	Ontology	Any type	High	-	Application based	✓
[50]	COSMOS	Distributed	Object Oriented	Physical	High	✓	Middleware based	-
[51]	CoBrA	Component	Ontology	Any type	High	-	Application based	✓
[52]	Hydra	Distributed	Ontology and Object	Physical	High	-	Application based	✓
[53]	COSAR	Stand-Alone	Ontology	Mobile	Low	-	Application based	✓
[54]	Ubiware	Stand-Alone	Ontology	Any type	High	✓	Middleware based	✓
[55]	CASS	Centralized	Logic and Object Oriented	Mobile	High	-	Application based	✓
[56]	QoS DREAM	Component	Logic and Object Oriented	Mobile	Low	✓	Middleware based	-
[57]	TinyRest	Centralized	Mark-up	Any type	High	-	Application based	-

Exercício

- 1 - Alguns elementos ou camadas são recorrentes na maioria das arquiteturas existentes para o suporte à execução de sistemas ubíquos e sensíveis ao contexto. Dentre esses elementos encontramos:
 - Camada de Sensoriamento;
 - Camada de Gerenciamento de Contexto;
 - Camada de Registro e Descoberta de Serviços;
 - Camada de Adaptação;
 - Camada da Aplicação
- A) Baseado em pelo menos duas das arquiteturas apresentadas em sala, explique qual a descrição e o papel de cada uma dessas camadas.
- B) Na aplicação que você imagina para o projeto como estariam distribuídos esses módulos/camadas na arquitetura do seu projeto