

Introdução à Computação Móvel e Ubíqua



Markus Endler

Sala 503 RDC

endler@inf.puc-rio.br

www.inf.puc-rio.br/~endler/courses/Mobile

Por que estudar Comp. Móvel e Ubíqua?

- Quantidade de disp. portáteis (smartphones/tablets) já superou a de computadores fixos;
- Grande maioria das aplicações são projetadas para acesso mobile (*mobile first*);
- Muitas aplicações usam a localização e dependem de sensores dos dispositivos;
- Cada vez mais, aparelhos/coisas do nosso dia-a-dia tem poder de processamento e interfaces wireless;
- Processamento embarcado em rede (networked embedded systems) é usado em muitas aplicações;
 - Monitoramento e controle autônomo de processos no mundo físico
- Estão surgindo padrões de fato para comunicação D2D wireless: ZigBee, MQTT, CoAP

Sistemas móveis e ubíquos

Características:

- Dinamicidade: devido aos recursos variáveis, e a mobilidade dos usuários
- Heterogeneidade (de HW e SW) dos recursos/dispositivos e redes wireless;
- Ausência de controle central: interação P2P, ad-hoc, temporária e oportunística entre dispositivos de usuários e recursos do ambiente;
- Larga escala (nr de usuários, de sensores e atuadores embarcados)
- Adaptação aos contextos (de sistema, físico e/ou social)
- Diferentes requisitos de QoS por parte das aplicações que compartilham os recursos;
- ...

Tais sistemas incluem vários desafios no desenvolvimento de software: roteamento, interoperabilidade, análise de informações de sensores, modelagem do contexto, etc.

Enfoque



- A disciplina **não** dará ênfase em:
 - ◆ Tecnologias wireless e protocolos de rede,
 - ◆ Desenvolvimento de aplicativos mobile,
 - ◆ Aspectos de interface com o usuário (HCI),
 - ◆ Segurança e privacidade.

- Em vez disto, iremos estudar:
 - ◆ desafios de sistemas móveis e ubíquos,
 - ◆ conceitos e princípios fundamentais para desenvolvimento de software
 - ◆ middlewares e aplicações.

Assuntos



- ◆ Introdução: Tipos de Rede, Aplicações, etc.
- ◆ Roteamento e Mobilidade
- ◆ Modelos de Comunicação e Coordenação
- ◆ Ciência do Contexto (context-awareness) e Modelagem de Contexto
- ◆ Serviços baseados em localização
- ◆ Descoberta de Serviços
- ◆ Exemplos de sistemas de middleware
- ◆ Processamento de fluxos de dados de sensores
- ◆ Smart Spaces
- ◆ Abordagens para a Internet of Things (IoT)
- ◆ Estudos de Caso de Sistemas (*Seminários por vocês*)

Avaliação

- Dois trabalhos práticos:
 - ◆ T1: protótipo de aplicação para comunicação instantânea que explora a mobilidade
 - ◆ T2: protótipo de aplicação que utiliza sensores no ambiente
 - ◆ Apresentação (demo) dos programas
 - * T1: em meados Outubro;
 - * T2: em meados Dezembro
- Pesquisa sobre middlewares para certo tipo de aplicação/rede
 - ◆ Monografia (12-15 páginas) meados Dezembro
 - ◆ Apresentação de seminários durante Dezembro
- Participação: Leitura e discussão de artigos em classe

Média final= (T1+ T2 + Monografia + Sem.+ Part.)/5

Trabalhos Práticos



Programação cliente-servidor, usando:

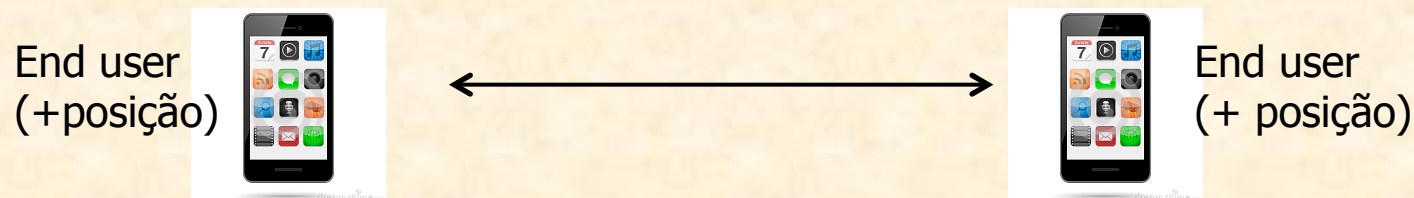
- middleware **Scalabe Data Distribution Layer (SDDL)**
 - ◆ Pub/Sub para dispositivos móveis (java/Android)
 - ◆ **Mobile Hub**: componente cliente que interage com sensores externos
- análise de dados de sensores usando Complex Event Processing
- Testes com SensorTags (interface Bluetooth 4.0 Low Energy)

Documentação, Tutoriais e código fonte do SDDL:

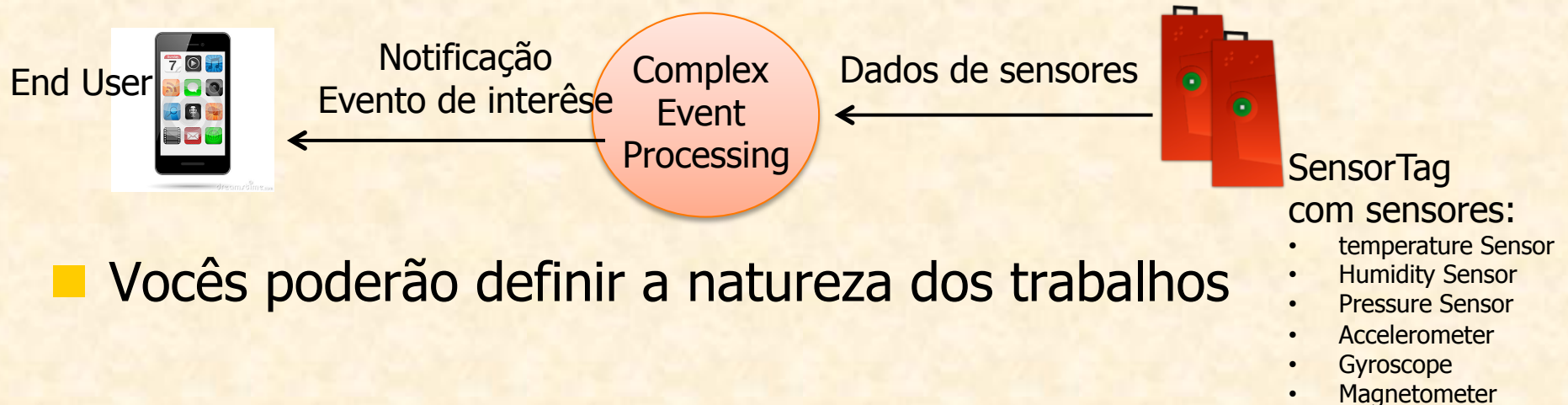
<http://www.lac-rio.com/dokuwiki/doku.php>

Trabalhos práticos

- T1: Comunicação instantânea, onde informação de localização é usada



- T2: Análise de fluxos de dados de sensores



- Vocês poderão definir a natureza dos trabalhos

➤ Scalabe Data Distribution Layer Middleware

- Motivation
- Main Challenges
- Enabling Technologies



Scalabe Data Distribution Layer Middleware

Project ContextNet aims at provisioning context services for wide- and large-scale pervasive collaborative applications such as on-line monitoring or coordination of mobile entities' activities, and information sharing through social networks. These entities may be users of portable devices (e.g. smartphones), vehicles, or autonomic mobile roboters. In the ContextNet project, all communication and context distribution capabilities are implemented in the **Scalabe Data Distribution Layer (SDDL)**, which is subject of this tutorial.



Download



Tutorial



SDDL architecture.

Motivation

Applications such as vehicle fleet monitoring and dispatch systems, logistics, emergency response coordination, or mobile workforce management, employ mobile networks as means of enabling communication, data sharing and coordination among a possibly very large set of mobile nodes – that be vehicles, people or event mobile robots and UAVs. The majority of those applications thus requires the capacity of handling real-time dissemination of context/ location information, group communication and management for thousands of mobile nodes, adaptability in very dynamic scenarios, where mobile nodes experience intermittent connectivity, or change of their IP address, and dynamic grouping of nodes according to their current location or any other mobile context information.

A common characteristics of all such applications is the fact that the mobile nodes periodically produce some data about them (e.g. probing sensors), which we call context information, like for example, their position, speed or other data, and publish this data to other nodes – either stationary or mobile, for being processed or visualized in almost real-time. Hence, it is assumed also that each mobile node has some wireless interface and is capable of communicating with other stationary machines, simply through the IP protocol. For all these applications, the main requirement is that, if the mobile node is connected and is producing its context data, this context update should be delivered to all the other interested nodes almost instantaneously.

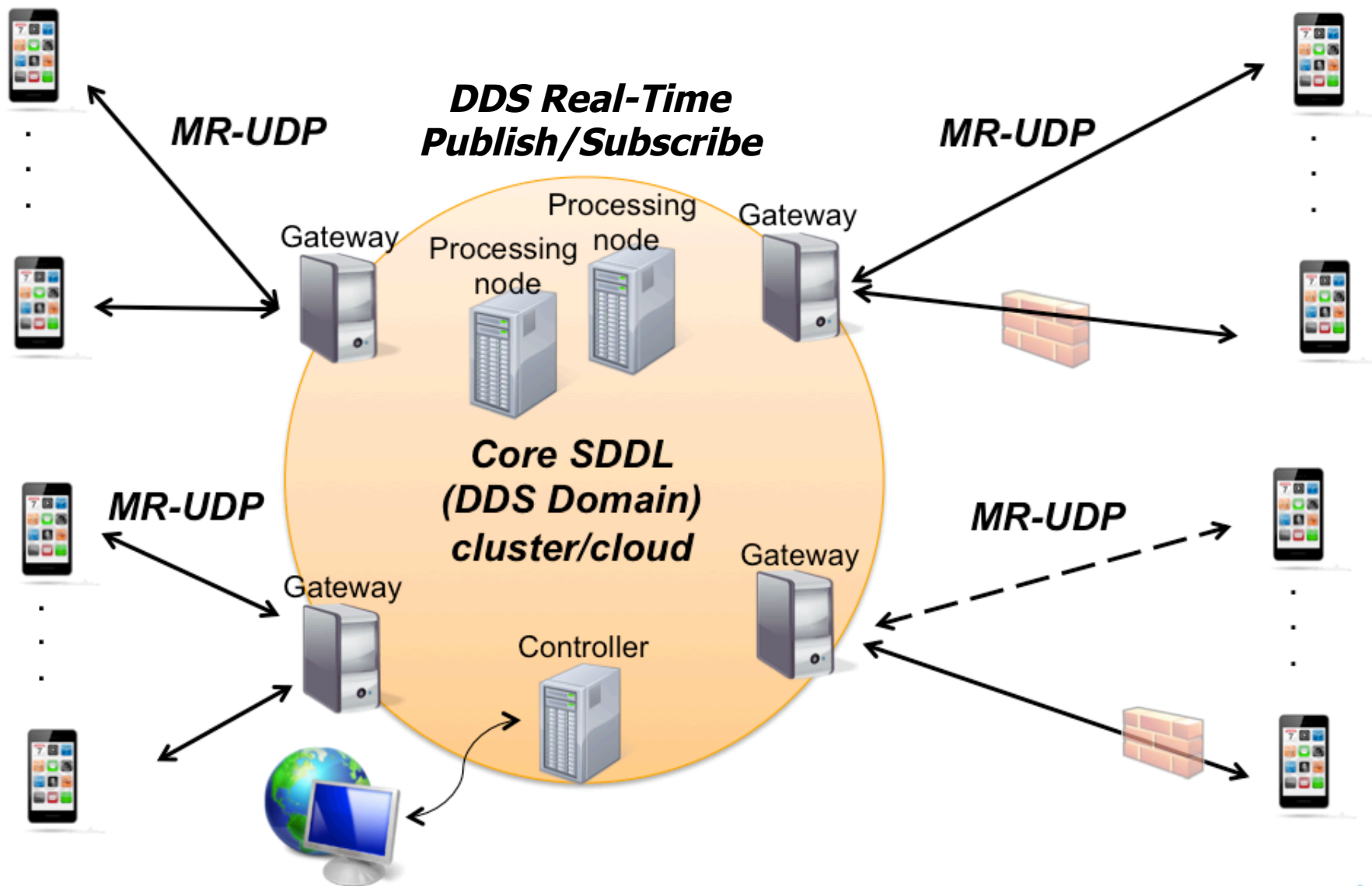
Project ContextNet builds on previous experiences and tools developed in Project Mobilis. It represents a general and long-term research initiative with several concrete projects as its spin-off activities.

Main Challenges

- Enable scalable distribution of context information among hundreds of thousands of context-producing and context-consuming entitie (user, vehicles, smart objects);
- Devise reasoning techniques that are inherently distributed and capable of detecting application relevant patterns of global context situations;
- Use semantic Web to combine several types of context information (computing, physical, time, user context) and integrate them with social networks to leverage the communication and coordination capabilities of mobile users and/or vehicles.

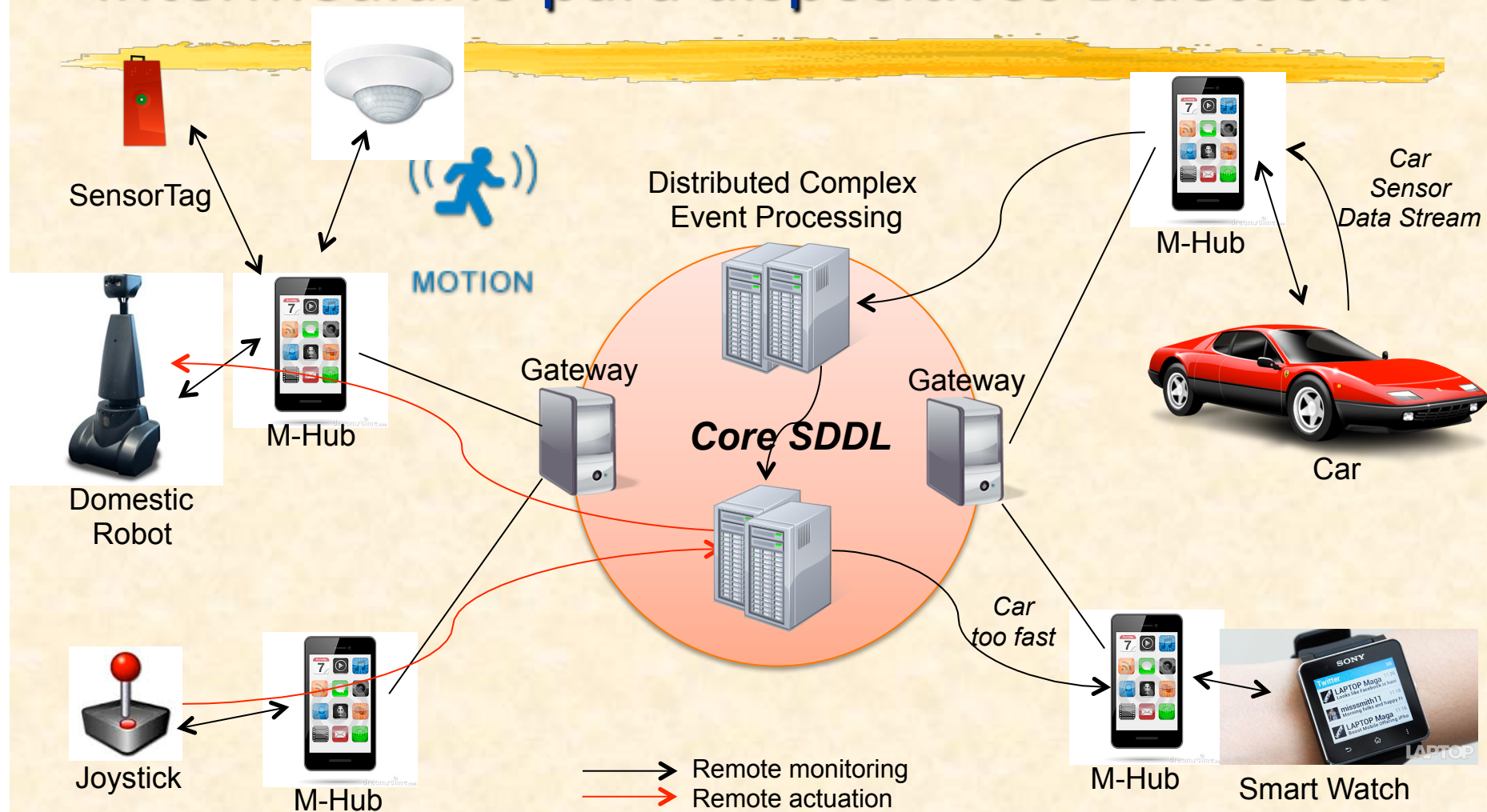
<http://www.lac-rio.com/dokuwiki/doku.php>

Visão Geral - SDDL

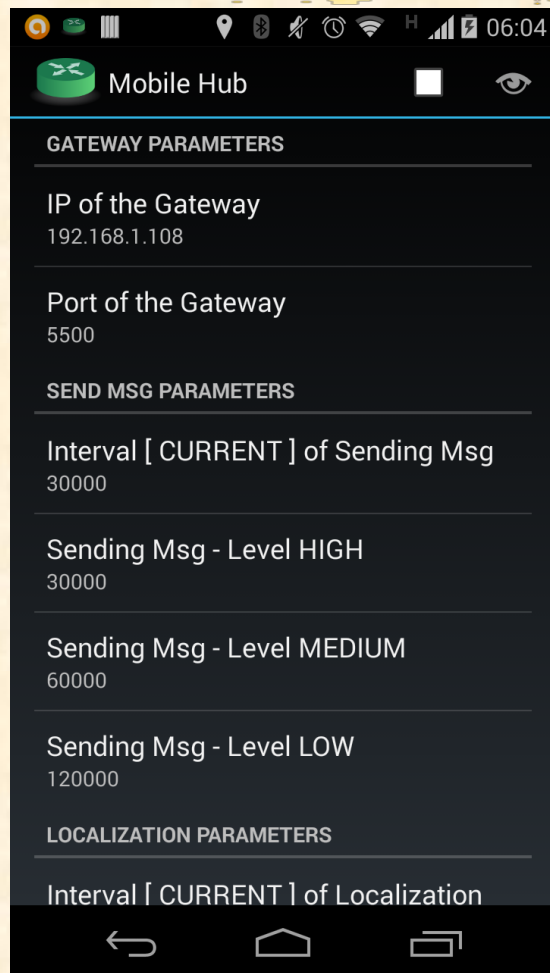


Mobile Hub

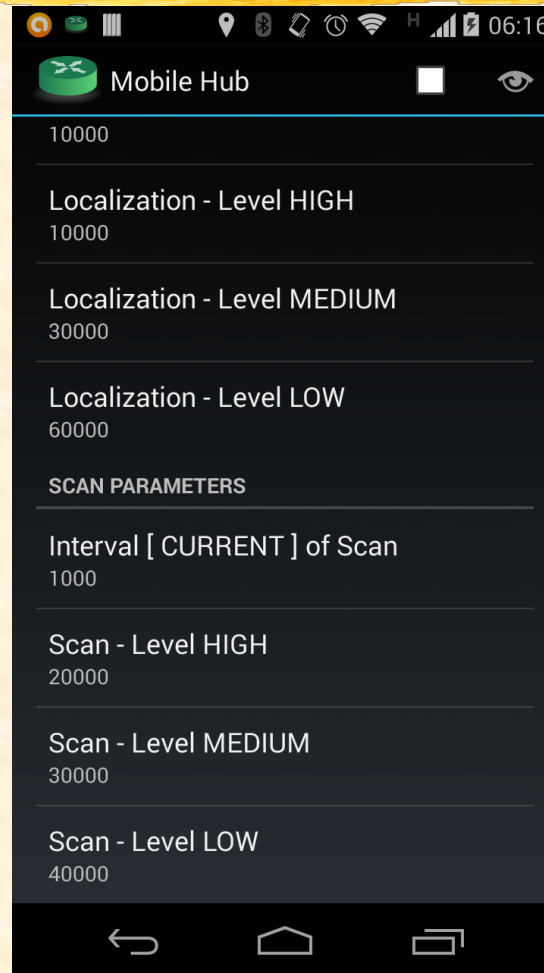
Intermediário para dispositivos Bluetooth



M-Hub 1.0 Viewer App



Internet parameters



Localization parameters



Nearby M-OBs and sensor data

Complex Event Processing

- Tecnologia para processamento de Fluxos de Eventos em tempo real
- Eventos podem ser de diferentes tipos (e origens)
- A lógica completa do processamento é definida por uma Event Processing Network, composta de Event Processing Agents (EPA)
- Cada EPA analisa todos os eventos em uma janela de tempo, executa regras que implementam filtros, transformações e/ou verificam padrões de eventos
- Regras do Tipo Event-Condition-Action
- EPAs podem chamar listeners ou produzir novos tipos de eventos (simples ou complexos)

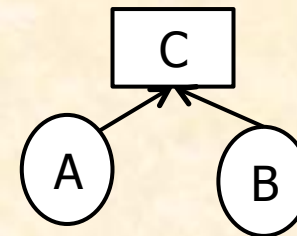
CEP Event Pattern (ECA) Example:

When:

"eventA -> eventB ("->" means followed by)
AND
Average(eventA.value).win.sec(10 sec) > value"

Generate Complex Event:

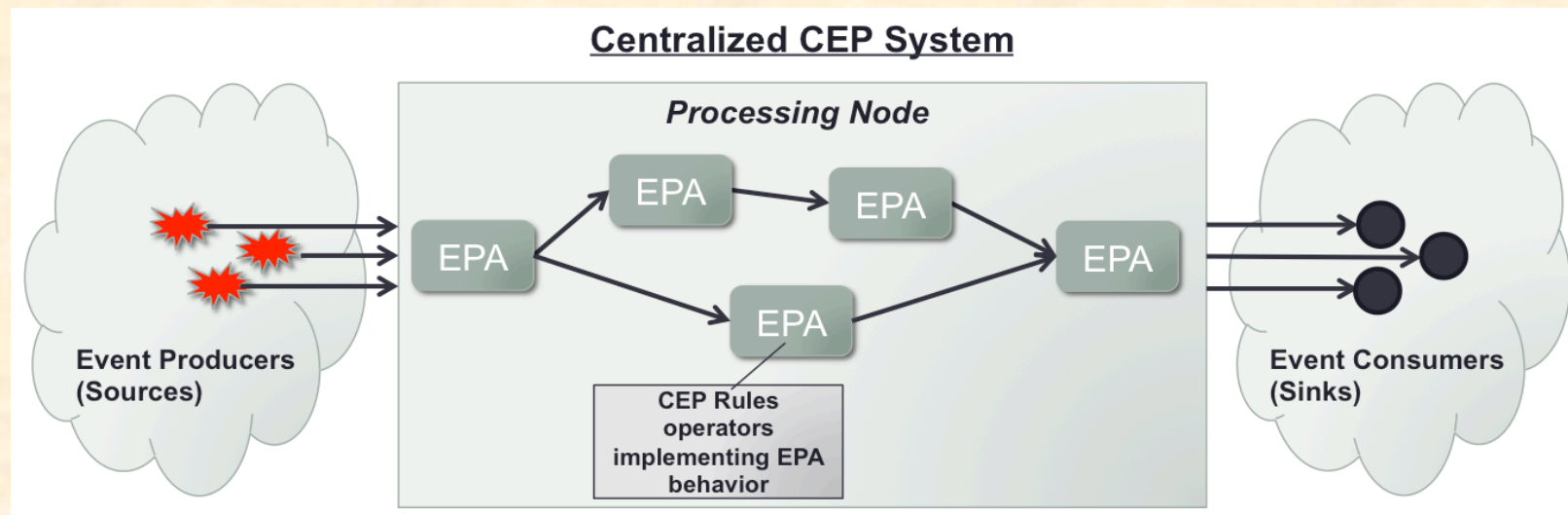
eventC (eventA, eventB)



EPAs e EPNs

Event Processing Networks (EPNs)

- Uma rede de EPAs interconectados
- Implementa a lógica global de processamento para a detecção de uma situação
- EPAs repassam ou geram eventos que são consumidos por outras EPAs



Esper



- Existem muitas engines para CEP
- Esper é uma ferramenta open-source (esper.codehouse.org) para Java e C#
- Consultas e regras escritas em linguagem declarativa EPL (event procesing Language), similar ao SQL
- O Mobile Hub contém uma engine Esper para Android

Monografia/Seminário

Título:

"Desafios e Soluções de Middleware para <classe de aplicação móvel ubíqua>"

Exemplos:

- ◆ sistemas embarcados distribuídos,
- ◆ comunicação entre veículos,
- ◆ Sensoriamento participativo (Crowdsensing),
- ◆ Redes de sensores sem fio,
- ◆ robótica distribuída,
- ◆ Home automation;
- ◆ Redes Sociais Pervasivas,
- ◆ Wearable computing
- ◆ ...

- Análise comparativa de sistemas de middleware (pelo menos 3-4) com relação aspectos, tais como:

paradigma de programação, ciência do contexto, escalabilidade, QoS, tempo real, interoperabilidade, dependabilidade, usabilidade (aplicações desenvolvidas).