

# S. empotrados y ubicuos

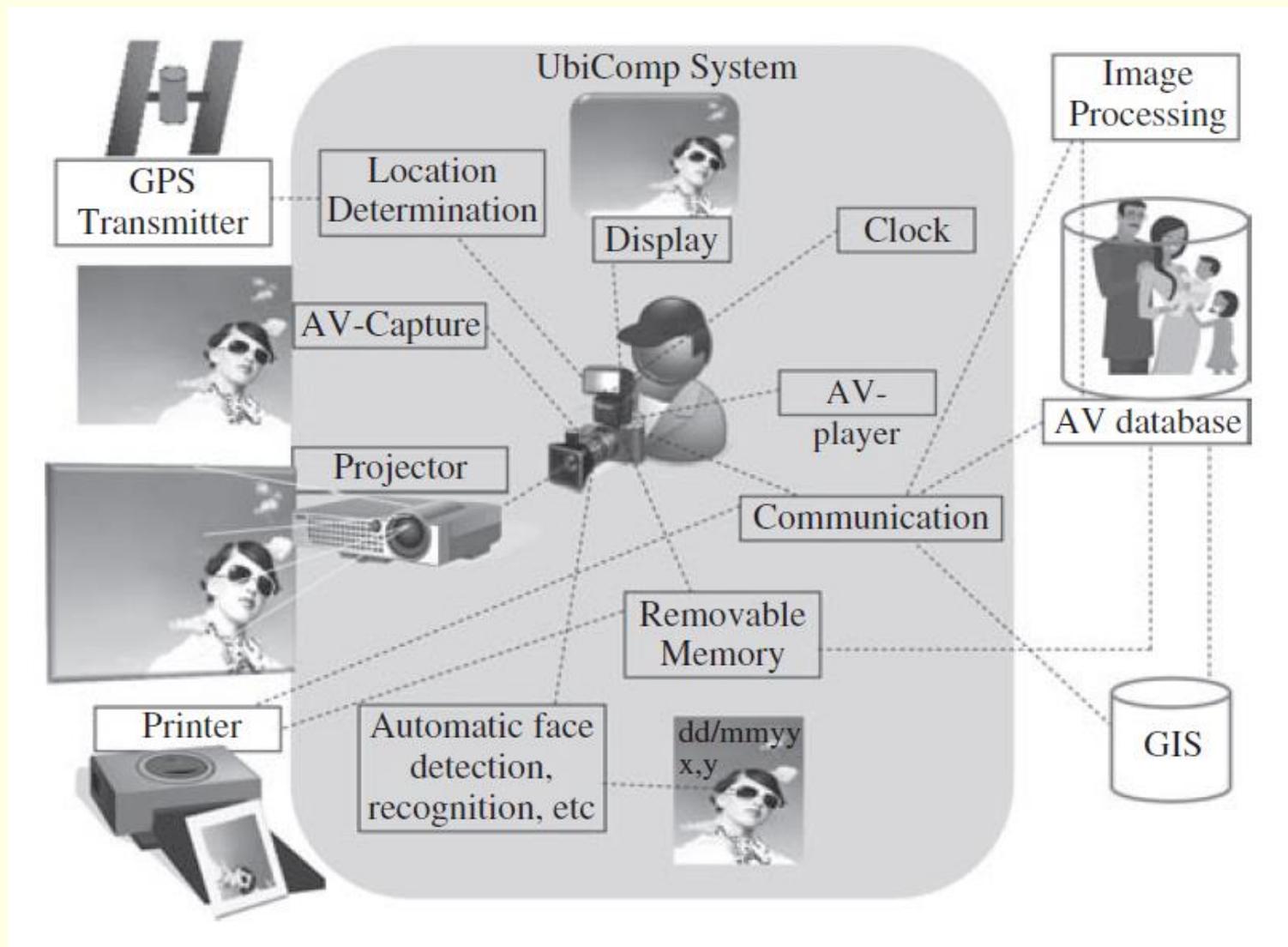
*Computación ubicua (UbiComp)*  
*Internet of Things (IoT)*

Fernando Pérez Costoya  
*fperez@fi.upm.es*

# Contenido

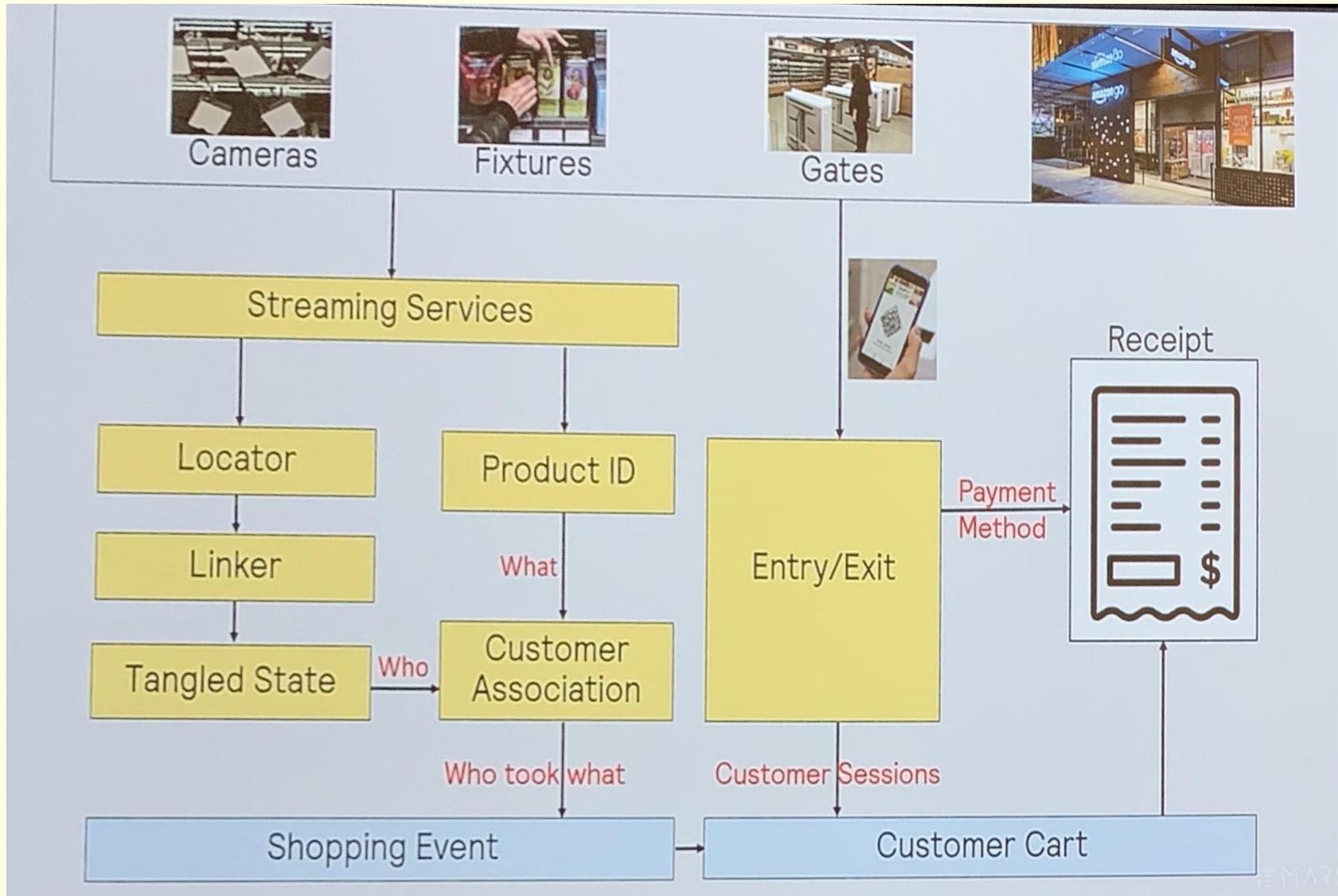
- ☐ Introducción a *UbiComp*
- ☐ Precedente: La computación móvil
- ☐ La visión de Mark Weiser
- ☐ Retos de la computación ubicua
- ☐ *Internet of Things* (IoT)
- ☐ Espacios inteligentes de interacción
- ☐ Arquitectura de los sistemas IoT
- ☐ *Context-aware Computing*
- ☐ Sistemas de localización
- ☐ *UUI*
- ☐ Seguridad y privacidad

# Dispositivo “fotográfico” UbiComp



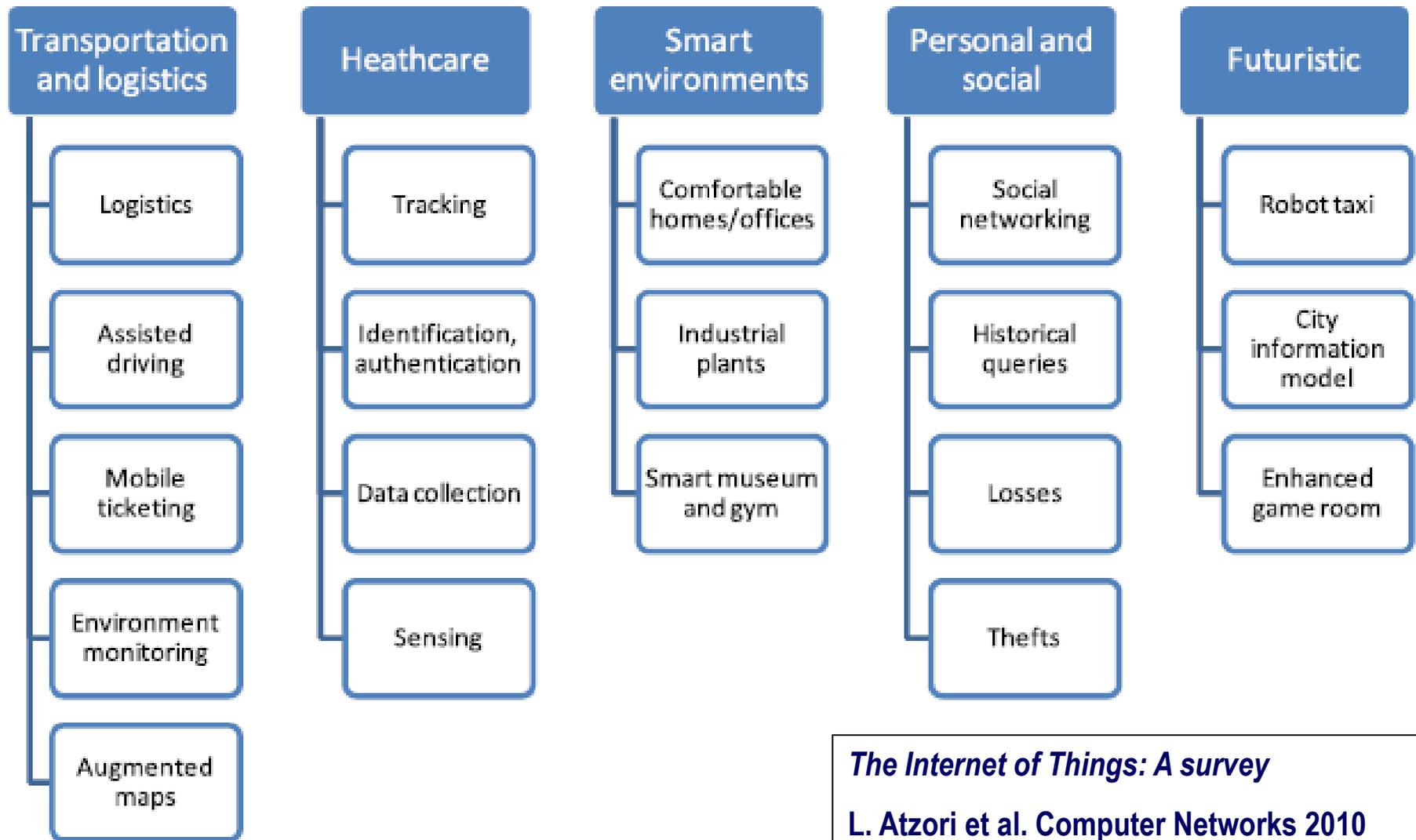
*Ubiquitous Computing: Smart Devices, Environments and Interactions. Stefan Poslad. 2009*

# Amazon Go: “Just Walk Out” experience



[towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9](https://towardsdatascience.com/how-the-amazon-go-store-works-a-deep-dive-3fde9d9939e9)

# Aplicaciones *UbiComp/IoT*



*The Internet of Things: A survey*  
L. Atzori et al. *Computer Networks* 2010

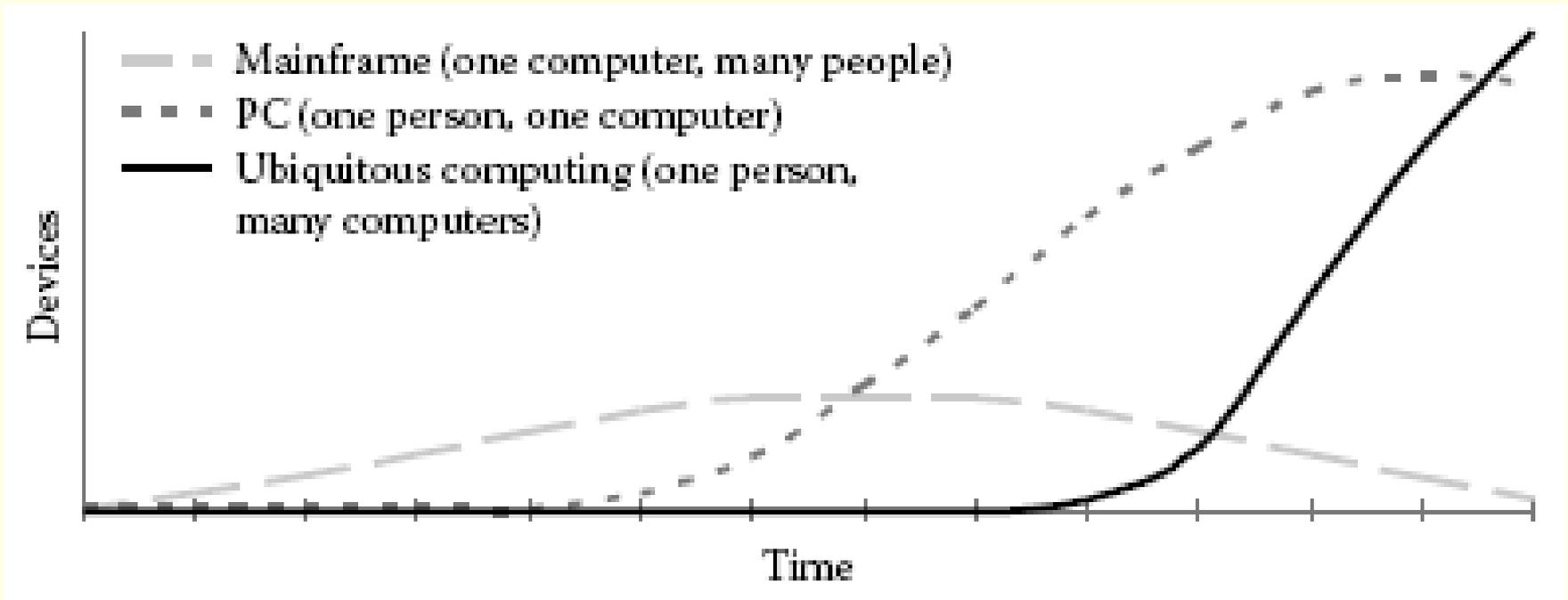
# Computación ubicua (*Ubiquitous Computing*)

- Tendencia: Progresiva miniaturización de computadores
  - Computadores empotrados en todo tipo de objetos
- Tendencia: Continua mejora en comunicación inalámbrica
  - Esos computadores empotrados interaccionan entre sí
- Nuevo modelo de computación:
  - Computadores omnipresentes, parte de ellos móviles
  - Algunos empotrados en sistemas físicos, invisibles al usuario
  - Otros portados, consciente o inconscientemente, por el usuario
  - Conectados entre sí ofreciendo un valor añadido
  - Plenamente integrados en el mundo para facilitar vida cotidiana
    - Aunque, probablemente, con menos **privacidad** y **sostenibilidad**
- **Computación ubicua (*UbiComp*)**
  - Propuesta de Mark Weiser en PARC de Xerox finales de los 80
    - *Pervasive Computing*: Propuesta de IBM; finalmente sinónimo
- Nueva era en la historia de la computación moderna

# Cuatro eras en la computación

- *Mainframes*
- Sistemas distribuidos (desde principio década 80):
  - Gracias a PCs y mejoras en redes cableadas
  - Máquinas en red trabajando de forma coordinada
  - Tecnologías maduras
- Computación móvil (desde principio década 90)
  - Gracias a mejoras en portátiles y redes inalámbricas
  - Conectados a infraestructura cableada o en redes *ad hoc*
  - Información “*anywhere anytime*”
  - Investigación desde década 90; tecnologías en maduración
- Computación ubicua (actualmente)
  - Información “*everywhere everytime*”
  - Tecnologías en efervescente investigación

# Evolución de la computación



## Computación ubicua:

- 1 persona → N computadores
- 1 computador → N personas

*Ubiquitous Computing Fundamentals*

John Krumm

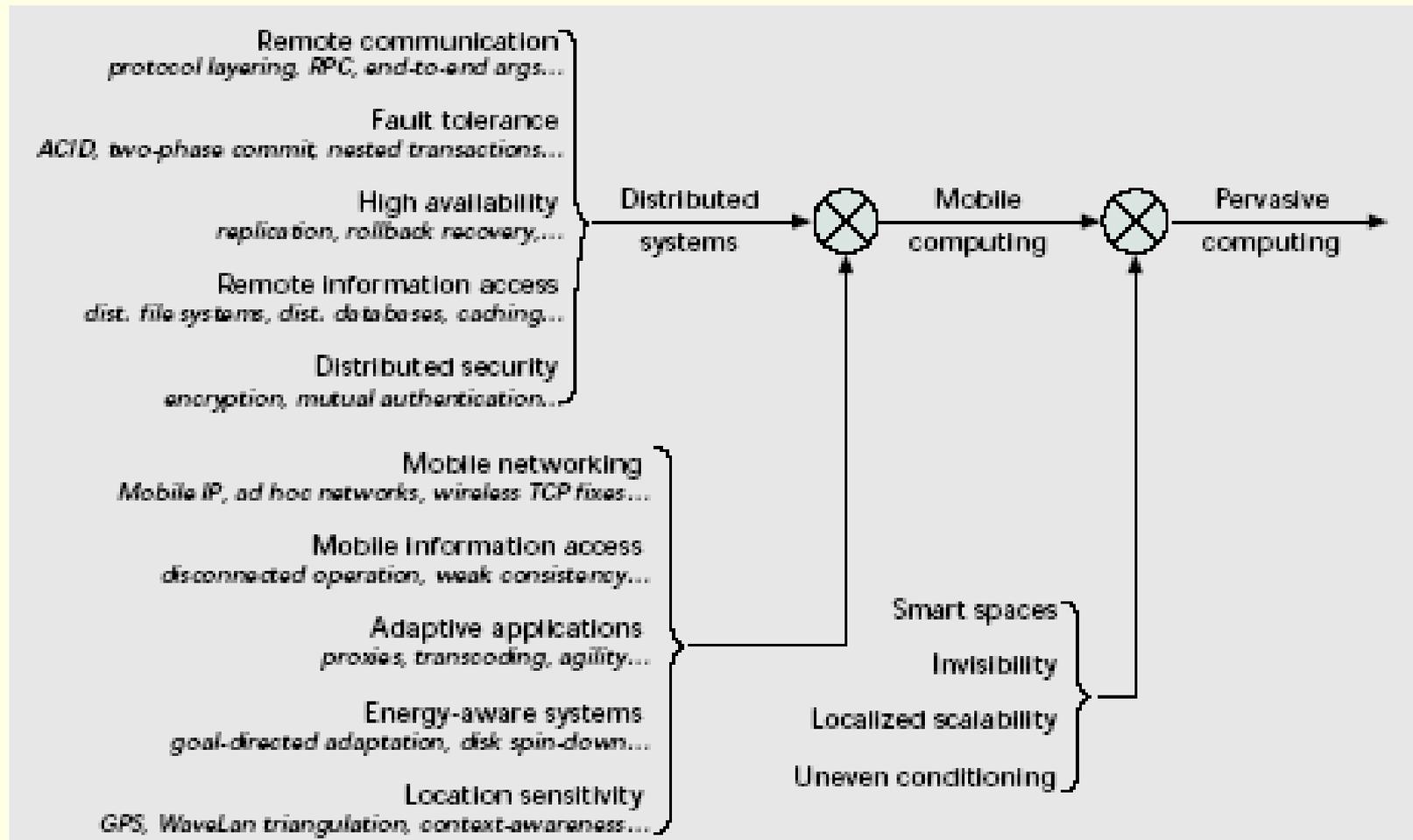
# Tipos de redes (extraído del libro de Coulouris)

---

|                  | <i>Example</i>                       | <i>Range</i> | <i>Bandwidth</i><br><i>(Mbps)</i> | <i>Latency</i><br><i>(ms)</i> |                                 |
|------------------|--------------------------------------|--------------|-----------------------------------|-------------------------------|---------------------------------|
| <i>Wired:</i>    |                                      |              |                                   |                               |                                 |
| LAN              | Ethernet                             | 1-2 kms      | 10-1000                           | 1-10                          |                                 |
| WAN              | IP routing                           | worldwide    | 0.010-600                         | 100-500                       |                                 |
| MAN              | ATM                                  | 250 kms      | 1-150                             | 10                            |                                 |
| Internetwork     | Internet                             | worldwide    | 0.5-600                           | 100-500                       |                                 |
| <i>Wireless:</i> |                                      |              |                                   |                               |                                 |
|                  | + Comunicación corto alcance: NFC... |              |                                   |                               |                                 |
| WPAN             | Bluetooth (802.15.1)                 | 10 - 30m     | 0.5-2                             | 5-20                          | WPAN: Crucial en <i>UbiComp</i> |
| WLAN             | WiFi (IEEE 802.11)                   | 0.15-1.5 km  | 2-54                              | 5-20                          |                                 |
| WMAN             | WiMAX (802.16)                       | 550 km       | 1.5-20                            | 5-20                          |                                 |
| WWAN             | GSM, 3G phone nets                   | worldwide    | 0.01-02                           | 100-500                       |                                 |

---

# De Sistemas Distribuidos a Computación ubicua



 **Complejidad se multiplica**

***Pervasive Computing: Vision and Challenges***

**M. Satyanarayanan**

# Retos de la computación móvil

- Sist. distribuidos → Computación móvil → Computación ubicua
  - Retos existentes más complejos y aparecen nuevos
- C. móvil comparte “problemas” con c. ubicua
  - Sistemas volátiles y con limitación en recursos disponibles
  - Pero exacerbados en computación ubicua
- Revisemos cómo se han afrontado los retos de la c. móvil por
  - Uso de comunicaciones inalámbricas
  - Movilidad de los equipos
  - Portabilidad de los equipos

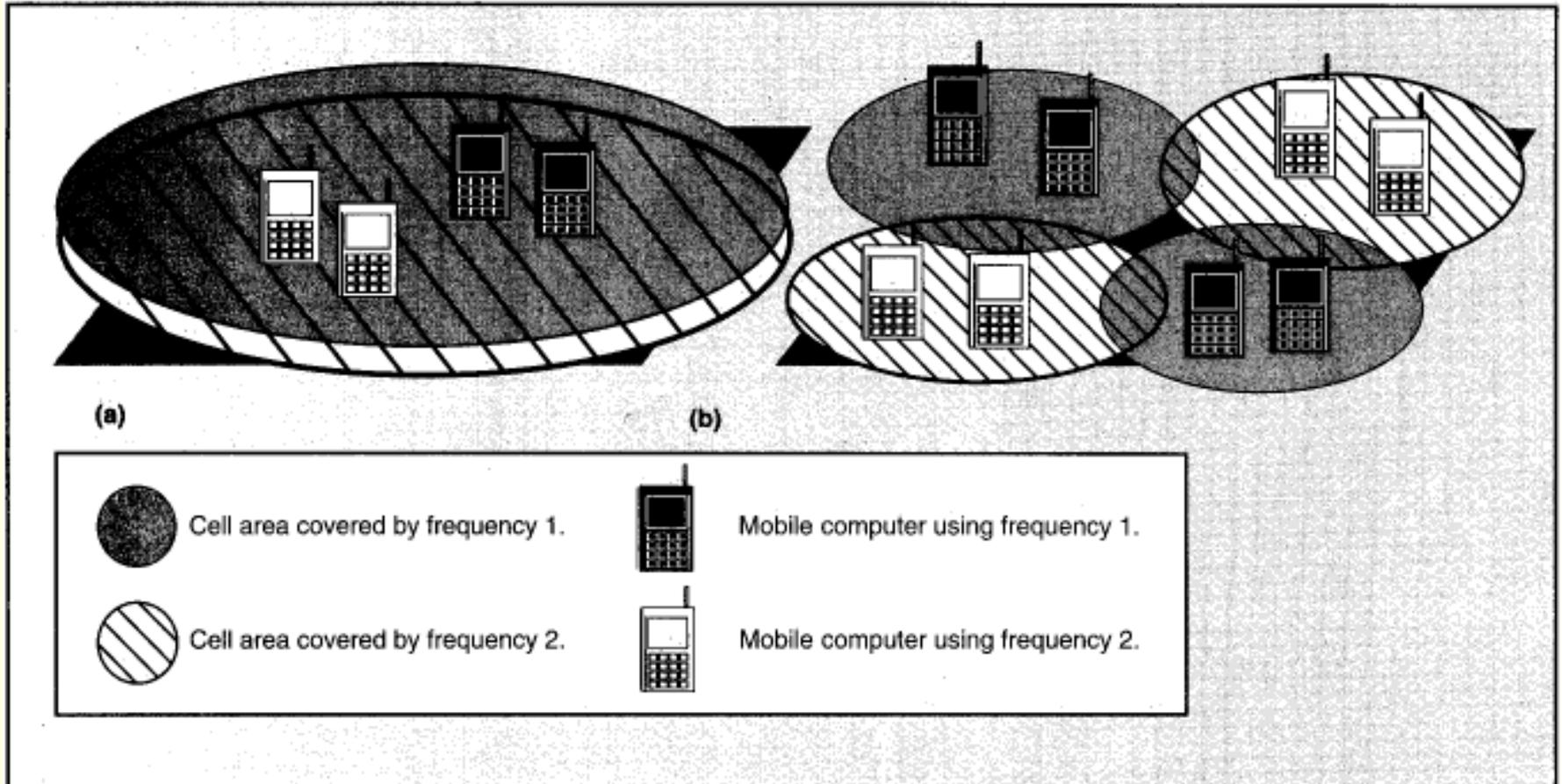
# Retos de c. móvil por comunicaciones inalámbricas

- Ancho de banda menor y más variable que en redes cableadas
  - Agrupar (*pre-fetching/delayed write*) y comprimir datos
  - **Adaptación** a ancho de banda disponible
    - P. e. Ajustar calidad de videoconferencia
    - Tipos de adaptación
      - ▶ Automática: SO y/o aplicación
      - ▶ Consultando a usuario
- Mayor propensión a errores y a la desconexión
  - Modo de operación desconectado
    - Ejemplo: Sistema de ficheros CODA
    - Requiere “*thick client*”: más recursos en clientes
      - ▶ “Fricción” entre autonomía y consumo de recursos
- Menos seguras: cifrado (adaptado a posible limitación de recursos)

# Rango de alcance y capacidad de la red

- ¿Cuánto mayor alcance mejor?
  - Intuitivamente, sí pero no es siempre así en c. móvil/*UbiComp*
- Limitar alcance de la comunicación:
  - Reduce consumo dispositivo móvil
  - Muy corto alcance posibilita comunicación directa entre nodos
    - P.ej. NFC
    - Puede posibilitar intercambios de datos seguros
  - Permite reutilización de ancho de banda
    - Más nodos en el mismo espacio físico
    - Concepto de bits/s/m<sup>3</sup>
      - ▶ ¿Cuántas comunicaciones simultáneas “cabén”?

# Rango de alcance y capacidad de la red

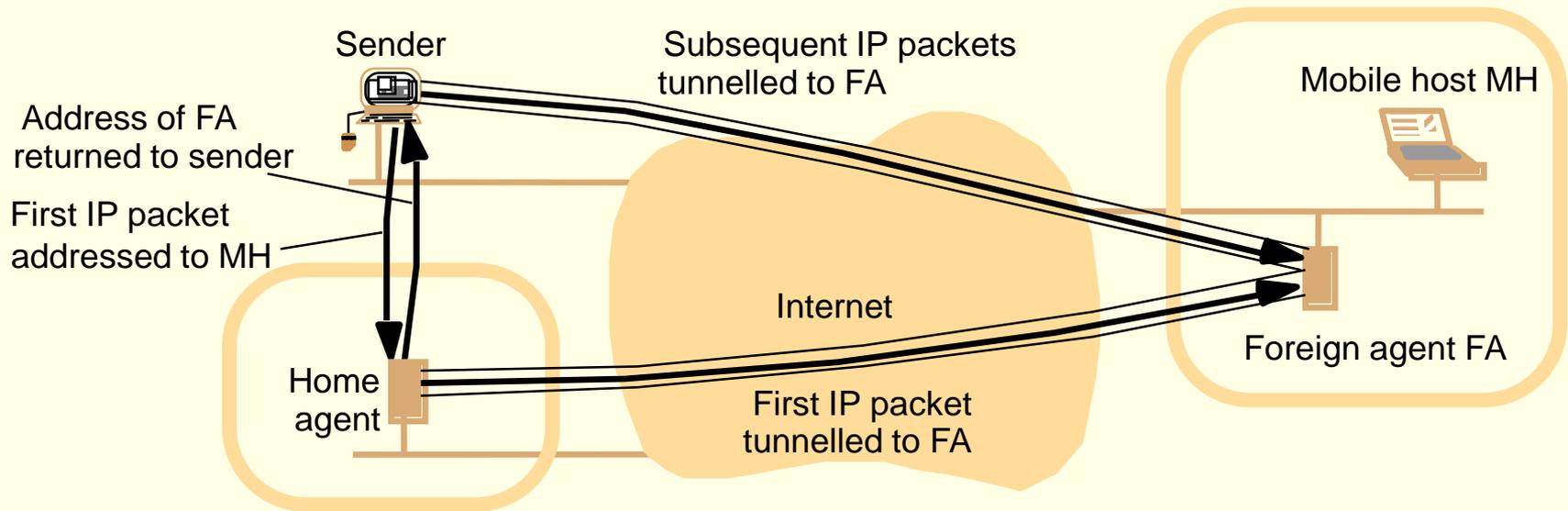


*The Challenges of Mobile Computing*  
G.H. Forman y John Zahorjan

# Retos de la computación móvil por movilidad

- Puede implicar cambio en tecnología de red usada
  - Variabilidad en ancho de banda
- Puede implicar cambio en punto de acceso a red cableada
  - Facilitar configuración automática
    - P. ej. DHCP, *dynamic DNS*, *Zero configuration*
  - Migración de direcciones
    - *Mobile-IP*
- Además de retos, abre nuevas posibilidades
  - *Location-based computing*
    - Comportamiento de aplicación depende de ubicación
    - Aspecto clave en computación ubicua
    - Implicaciones en aspectos de privacidad

# Mobile IP



***Distributed Systems: Concepts and Design***  
**Coulouris, Dollimore y Kindberg**

# Retos de la computación móvil por portabilidad

- Portátiles: recursos limitados (energía, UCP, memoria, HW de IU)
- Energía por baterías (alternativa: *Power Harvesting*)
  - HW de bajo consumo (p. e. UCP menor frecuencia y voltaje)
  - SW diseñado para reducir consumo
    - Algoritmos que premien bajo consumo sobre eficiencia o calidad
    - Esquemas con más cómputo y menos comunicación
      - ▶ P.e. sensor envía evento al cumplirse condiciones y no periódicamente
      - ▶ Con más recepción que envío en el portátil
    - Poner en bajo consumo recursos no usados
  - *Cyber foraging*:
    - Envío tareas complejas a máquinas de infraestructura red cableada
  - **Adaptación** a energía disponible (*energy-aware adaptation*)
    - P. e. Ajustar calidad de videoconferencia
  - *Resource-aware computing*
    - Aplicación notificada de nivel de disponibilidad recursos vitales

# Retos de la computación móvil por portabilidad

## ■ HW de IU limitado

- Pantalla de “menos calidad”
- Dispositivos de entrada alternativos: *pen*, voz, táctil, ...
- Objetivo: aplicaciones independientes del dispositivo
- **Adaptación** de la aplicación a:
  - Características del dispositivo
    - ▶ HW de IU (y limitaciones de recursos: energía, ancho de banda...)
  - Preferencias del usuario
- Generación de contenido adecuado para un dispositivo:
  - Estática:  $N$  versiones del contenido previamente generadas
  - Dinámica: por demanda
    - ▶ Se genera sobre la marcha contenido adaptado al dispositivo
    - ▶ Si contenido multimedia, operación de adaptación → *transcoding*

## ■ “Fragilidad”: golpes, pérdidas, robos, ...

# Computación ubicua: la Visión de Weiser

- Computadores deberían integrarse en nuestro entorno
  - Haciéndose “invisibles”, no requiriendo nuestra atención
    - *Invisible computing*
      - ▶ Como la escritura o el uso de la electricidad en el hogar
  - Facilitando contacto personal directo en vez de aislar
  - *Calm technology*: que se mantiene en la “periferia”
    - Puntualmente centro de nuestra atención y vuelve a periferia
- Lo opuesto a realidad virtual (realidad aumentada)
  - RV. Simula el mundo real: mundo → computador
  - CU. Mejora/“aumenta” el mundo real: mundo ← computador
- Prueba de concepto: CU en dispositivos que muestran información
  - Tres escalas: *tab* (nota), *pad* (bloc) y *board* (pizarra)
    - ▶ Pulgada, pie, yarda
    - En cada oficina 100-200 *tabs* , 10-20 *pads* , 1-2 *boards*
  - Actualmente, otras escalas: *dust*, *skin*, *clay*...

# *The Computer for the 21st Century, Weiser 1991*

*“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”*

*“Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence”*

*“We are therefore trying to conceive a new way of thinking about computers, one that takes into account the human world and allows the computers themselves to vanish into the background”*

*“There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods”*

# *Tabs*

- ☐ Actúa como terminal de la infraestructura de red
  - Muestra información recibida y envía acciones del usuario
- ☐ Puede “vestirlo” el usuario: *Wearable Computing*
  - *Active badge*
- ☐ Pantalla de contacto y 3 botones
- ☐ Comunicación inalámbrica IR
  - Menos consumo que RF: factor crítico en *Tabs*
- ☐ Detector IR en techo de oficina conectado a red cableada
- ☐ Capacidad de conocer su ubicación (oficina) en un edificio
  - *Tab* emite su ID mediante IR
- ☐ Posibilita aplicaciones *context-aware (location-aware)*
  - Desvío de llamada a oficina donde se encuentra destinatario
  - *Proximity Browser*
    - Ver ficheros consultados última vez en misma ubicación

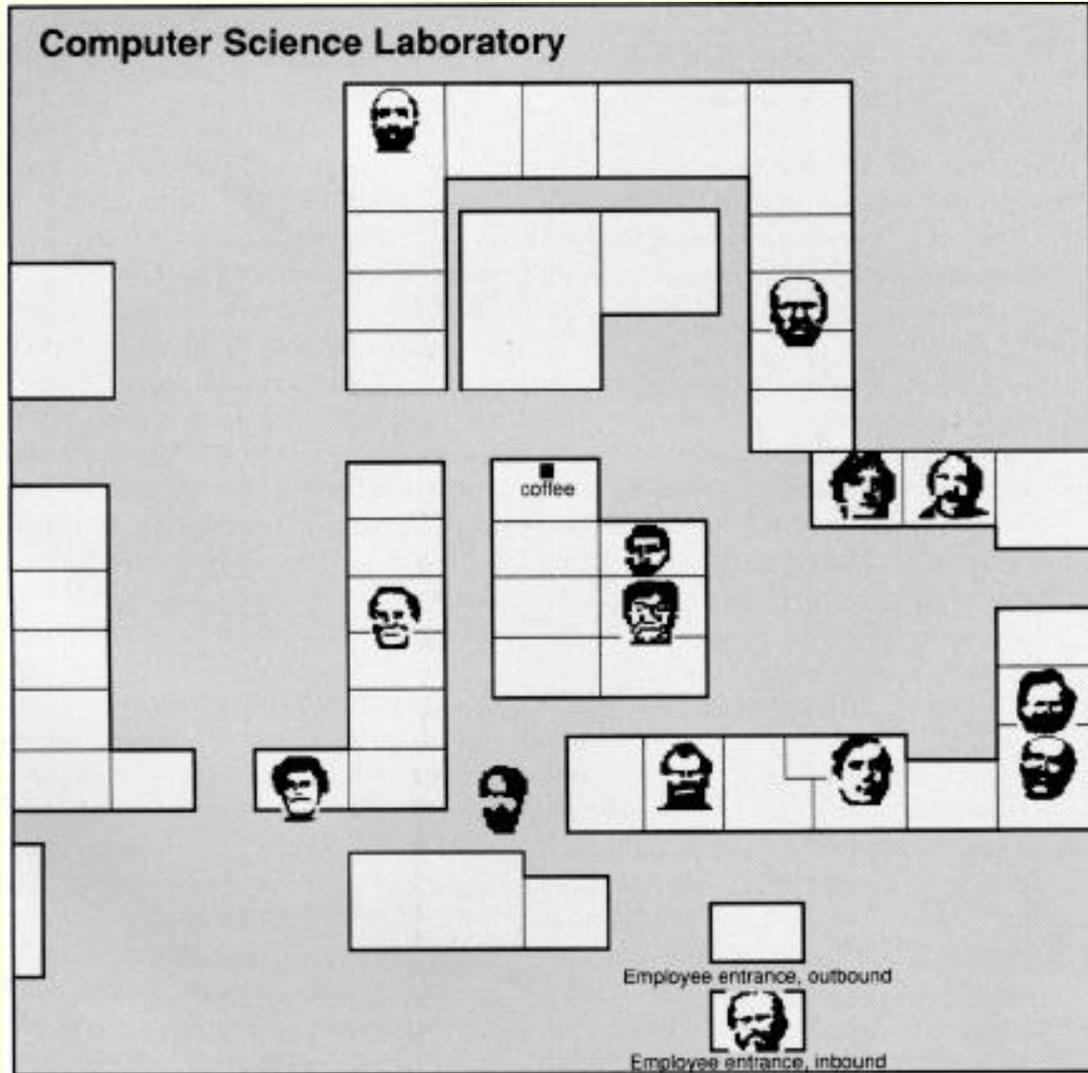
# *Tabs*



***Some CS issues in UC***  
**Mark Weiser**

***UC Fundamentals***  
**John Krumm**

# Seguimiento de ubicación mediante *tabs*

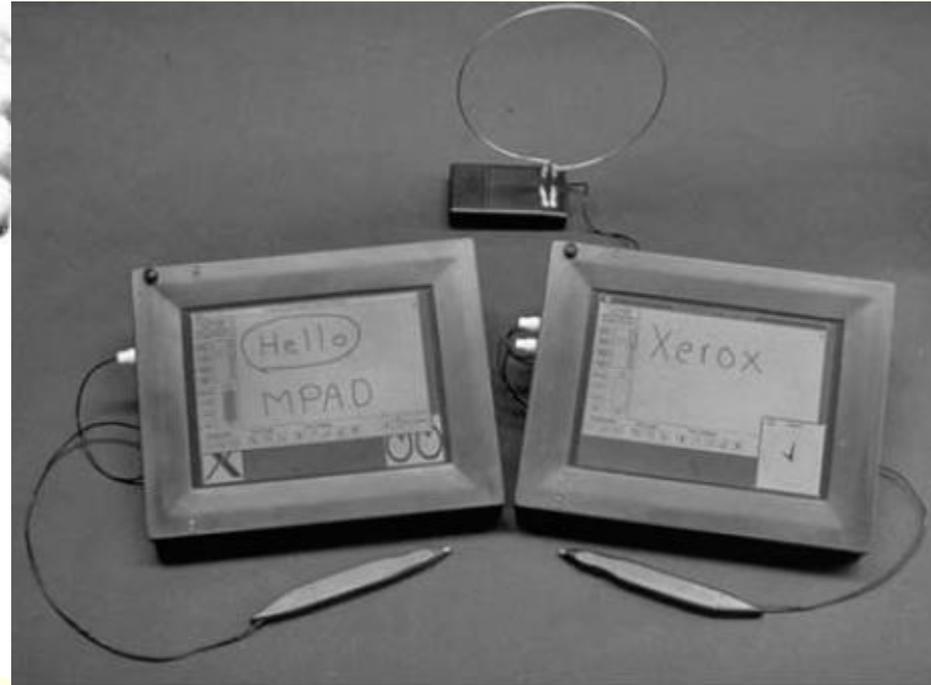
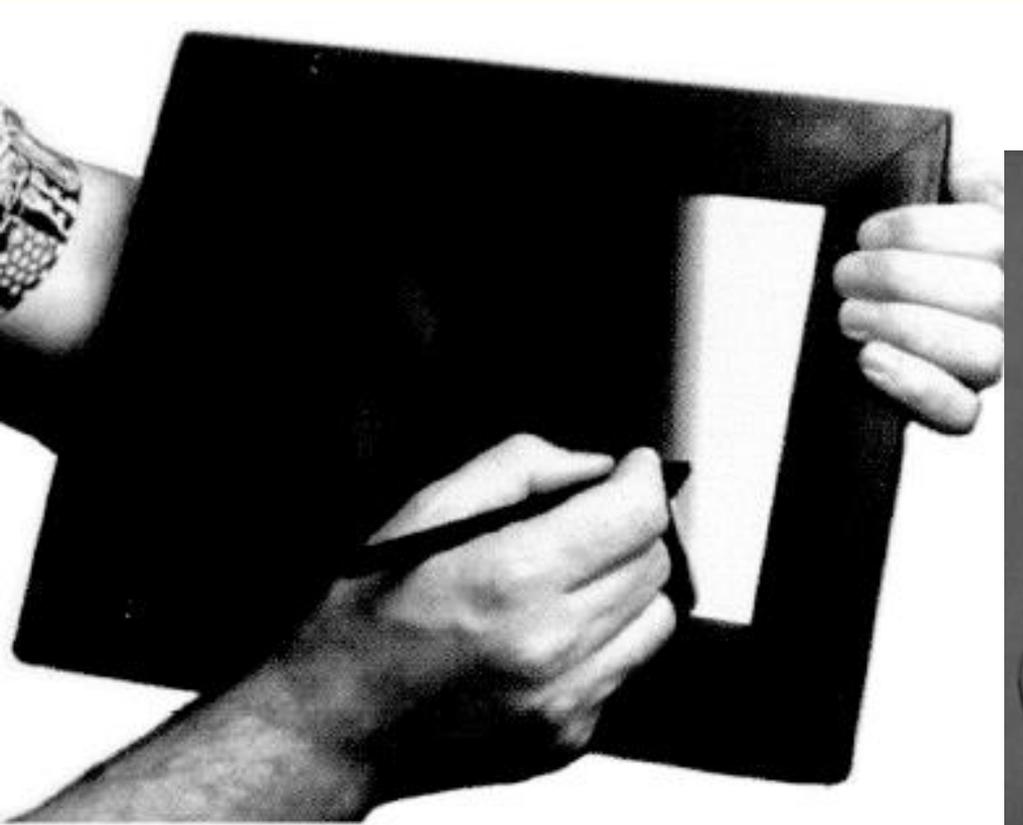


*Some CS issues in UC*  
Mark Weiser

# *Pads*

- Pantalla de visualización y escritura con bolígrafo
- Compatible con sistema de ventanas X
- Comunicación inalámbrica RF de corto alcance
  - Similar alcance que IR pero mucho mayor ancho de banda
  - Detector RF en techo de oficina conectado a red cableada
- Uso como “papel de borrador”
  - No dispositivos de uso personal: no es un portátil
  - Rol muy diferente a *tablets* actuales
- Metáfora en PC:
  - Pantalla → escritorio; Ventana → documento sobre escritorio
- *Pad* hace real la metáfora → documento sobre escritorio físico

# *Pads*



***Some CS issues in UC***

**Mark Weiser**

***UC Fundamentals***

**John Krumm**

# *Boards*

- Gran pantalla fija con entrada basada en bolígrafo IR
- Conectados directamente a infraestructura de red y a alimentación
- Actúan como tabloneros, pizarras, carteleros, ...
- Trabajo remoto en colaboración usando múltiples *boards*
- Ejemplo de uso: en docencia
  - Capturar anotaciones y gráficos del profesor durante la clase
  - Añadir info. de contexto para poder accederlos posteriormente
  - Gestión línea de tiempo para moverse a lo largo de presentación

# Boards



***Some CS issues in UC***

**Mark Weiser**

***UC Fundamentals***

**John Krumm**

# Otros trabajos pioneros

## ☐ *HP Cooltown*

- Precedente de *Web of Things*: todo objeto tiene una URL
  - *Web of Things*: “nivel de aplicación” de la *Internet of Things*

## ☐ MIT Media Labortary

- *Wearable Computing* y realidad aumentada
  - Precedente de Google Glass

## ☐ Georgia Tech: edificios inteligentes

- Classroom 2000, Aware Home

## ☐ Olivetti: sistemas de localización

- Active Badge, Active BAT

## ☐ Karlsruhe: *Things that Think*

- MediaCup, Smart-Its
  - <http://www.sciencedirect.com/science/article/pii/S1389128600001808>

## ☐ UC Berkeley: *Smart Dust*

- Redes de nodos sensores (*motes*) basados en MEMS

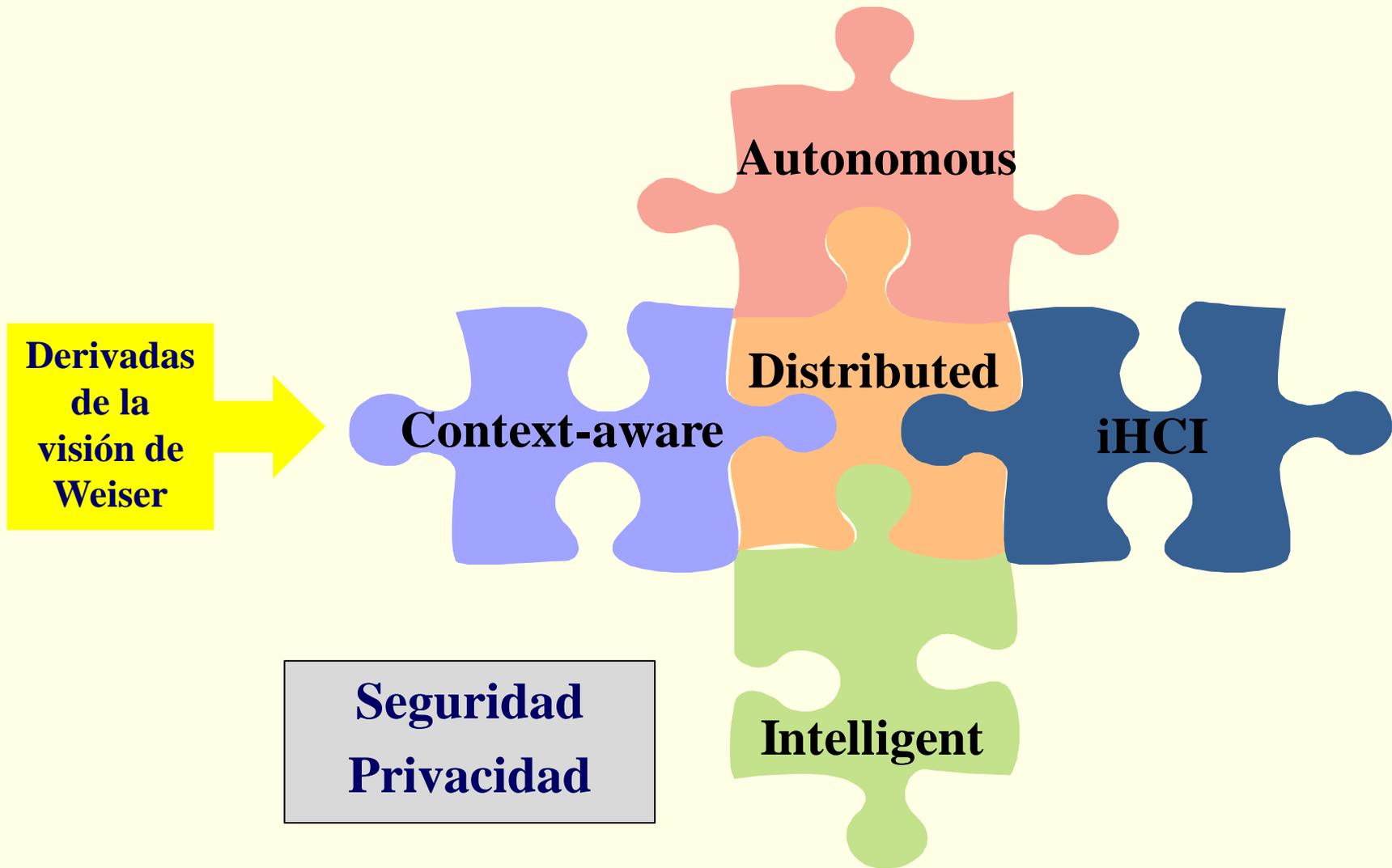
# Déficit tecnológicos detectados por Weiser

- ❑ Necesidad de computadores de bajo coste y consumo
- ❑ Mejoras en pantallas grandes y bolígrafos electrónicos
- ❑ Redes inalámbricas:
  - Mayor capacidad, mejores protocolos, distintos alcances...
  
- ❑ Más de 20 años después esos déficit parecen superados
- ❑ Algunas de las ideas se están incorporando a nuestra cotidianidad
- ❑ Sin embargo, CU no ha penetrado radicalmente en la sociedad
  
- ❑ ¿Qué retos quedan por resolver?

# Retos actuales de la computación ubicua

- Todos los de la computación móvil pero acentuados
  - HW con recursos más limitados (energía, UCP, HW de IU, ...)
    - Mayor necesidad de adaptación
  - Mayor escala por omnipresencia
  - Problemas de seguridad y privacidad exacerbados
- Gestión de espacios inteligentes
  - Sistemas volátiles con interacción espontánea
- Integración en el mundo físico: *Context-aware computing*
- Interfaces de usuario para sistemas ubicuos (*UUI*)

# Propiedades fundamentales de *UbiComp*



*Ubiquitous Computing: Smart Devices, Environments and Interactions.* Stefan Poslad. 2009

# Interacción entre “mundos” en *UbiComp*

- Mundo “humano” (H): personas y sus organizaciones
  - *H2H*: interacción entre personas
- Mundo “computacional” (C): equipos y comunicaciones
  - *C2C*: interacción entre equipos
- Mundo “físico” (P)
  - *P2P*: interacción entre elementos físicos
- Interacción entre personas y equipos
- Interacción entre equipos y componentes físicos
- Sistema convencional:
  - No interacción entre computadores y mundo físico
  - Usuario se adapta a interfaz de equipo (pantalla, teclado, ratón)
- Sistema *UbiComp* cambia el paradigma

# *H2H/H2C/C2H/C2C*

## ■ *H2H:*

- Personas interaccionan con mínimo soporte computacional

## ■ *H2C:*

- Persona se adapta a equipo para interaccionar
- *eHCI*: Interacción persona-computador **explícita**
  - p.e. pantalla, teclado, ratón

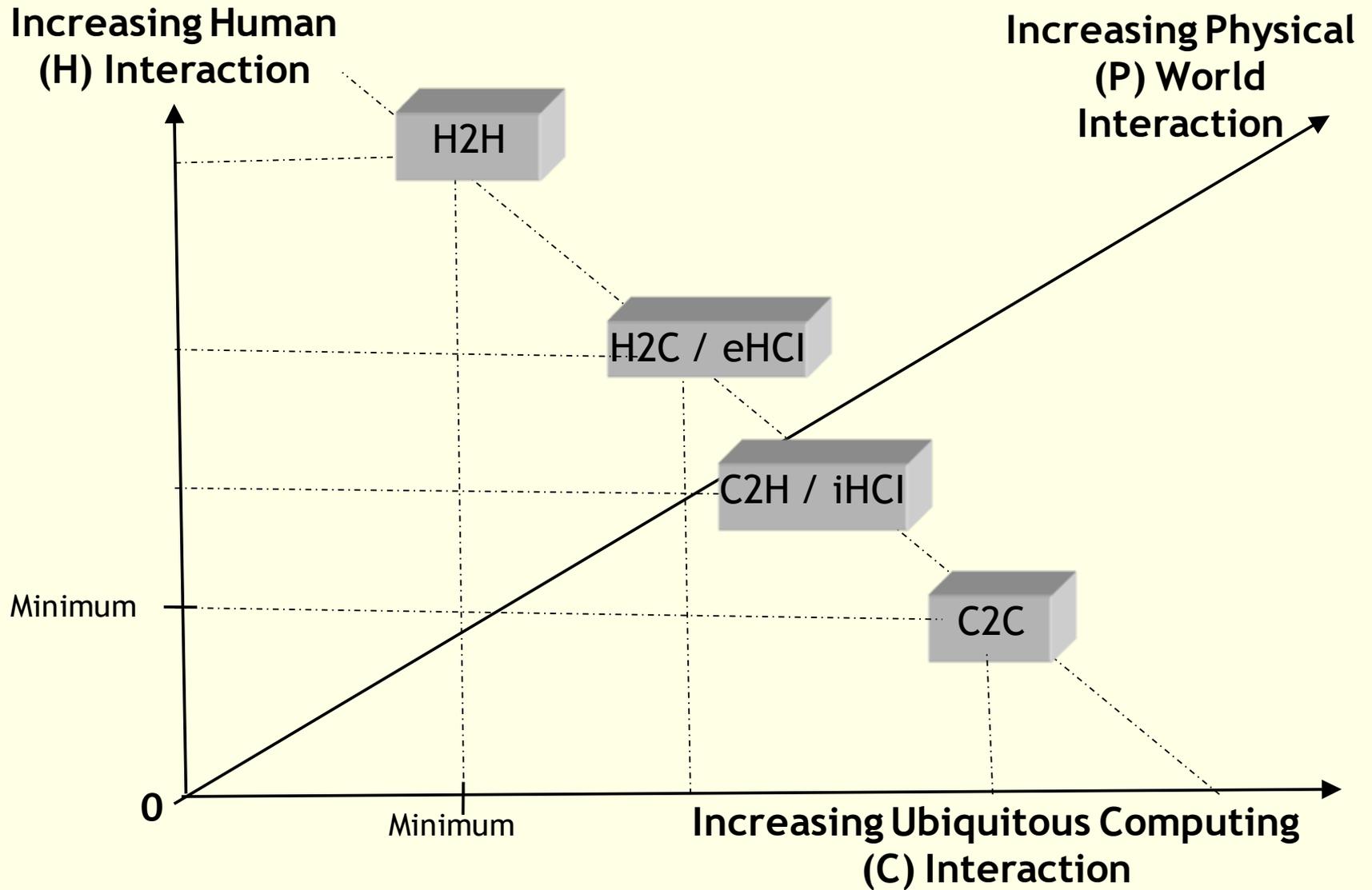
## ■ *C2H:*

- Equipo se adapta a persona para interaccionar
- *iHCI*: Interacción persona-computador **implícita**
  - p.e. lenguaje natural

## ■ *C2C:*

- Equipos interaccionan con mínimo soporte humano

# H2H/H2C/C2H/C2C



*Ubiquitous Computing: Smart Devices, Environments and Interactions. Stefan Poslad. 2009*

# *P2P/P2C/C2P/C2C*

## ☐ *P2P:*

- Elementos físicos interactúan sin soporte computacional

## ☐ *C2P:*

- *CA: (Physical) Context-aware*
- Equipo interactúa de forma simple con entorno físico
  - p.e. sensor detecta temperatura baja aumenta potencia calefacción

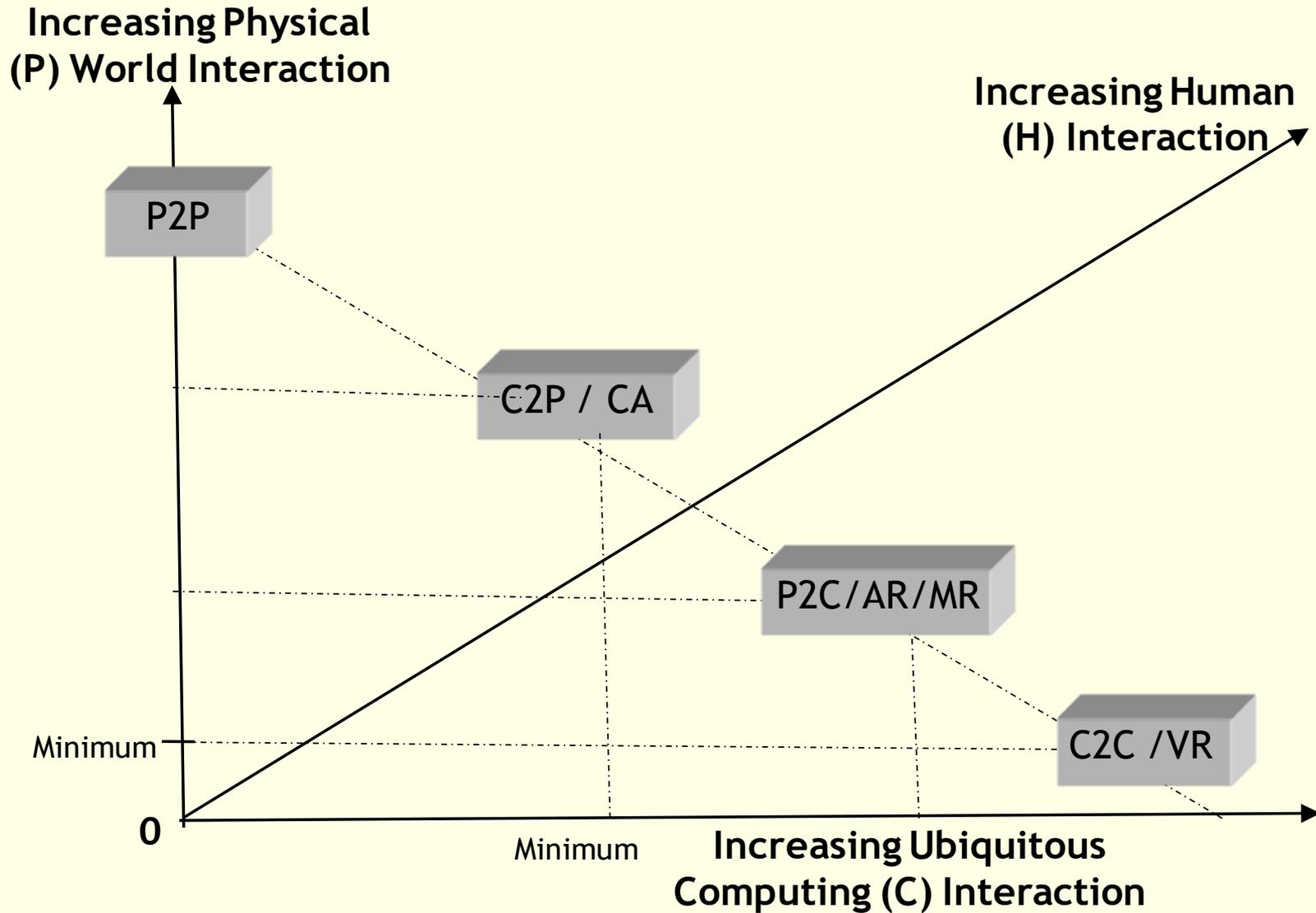
## ☐ *P2C:*

- *AR/MR: Augmented/Mediated Reality*
- Equipo incorpora el mundo físico a su computación
  - p.e. superposición de imágenes virtuales sobre reales

## ☐ *C2C:*

- *VR: Virtual Reality*
- Equipos con mínima interacción con entorno físico

# *P2P/P2C/C2P/C2C*



*Ubiquitous Computing: Smart Devices, Environments and Interactions. Stefan Poslad. 2009*

# Modelo *Smart DEI*

*Ubiquitous Computing: Smart Devices, Environments and Interactions.* Stefan Poslad. 2009

☐ *Smart Devices*

☐ *Smart Environments*

☐ *Smart Interactions*

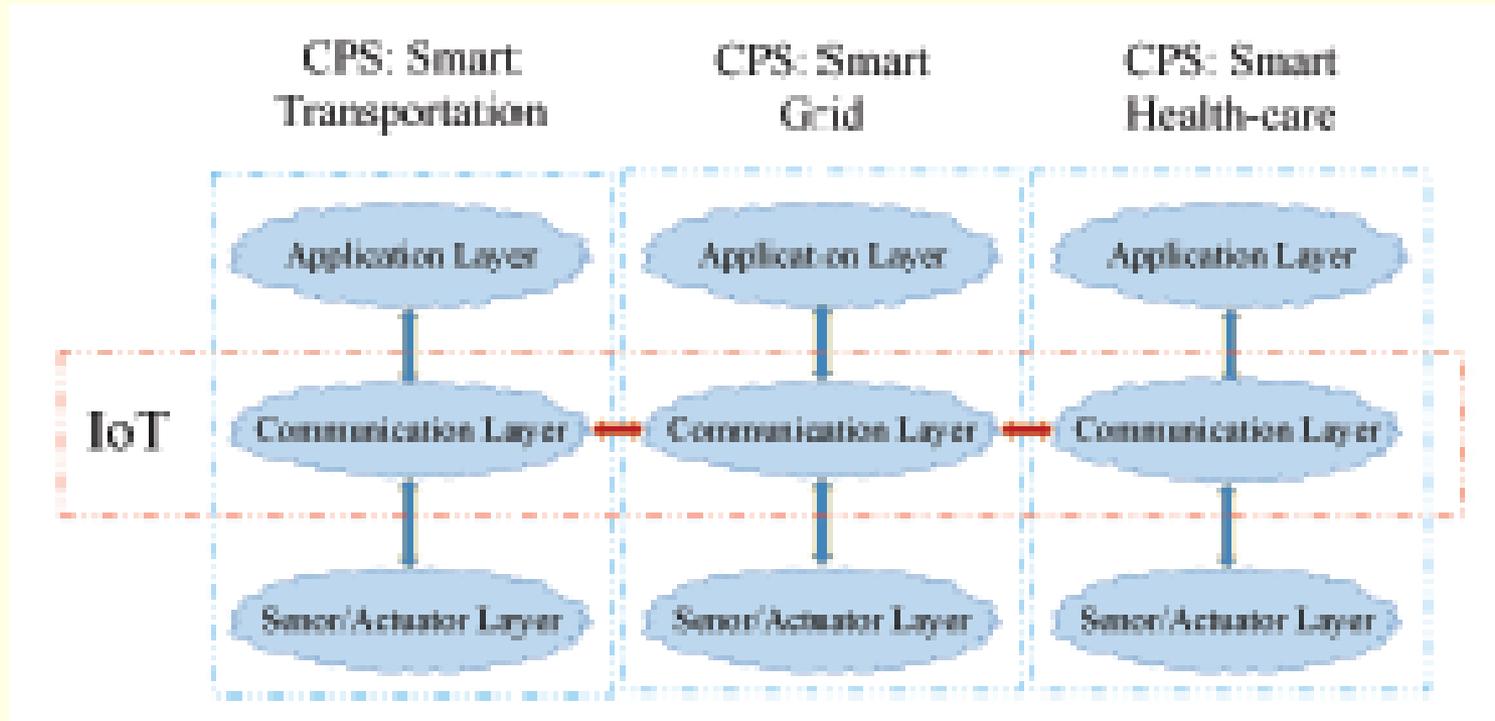
☐ Sistema *UbiComp*:

- Dispositivos inteligentes ubicados en entornos (espacios) inteligentes que interactúan de forma inteligente

# *Internet of Things (IoT)*

- ❑ Múltiples definiciones. Una sencilla del diccionario de Oxford:  
“*The interconnection via the **Internet** of computing devices embedded in everyday objects, enabling them to send and receive data.*”
- ❑ Término acuñado por Kevin Ashton en 1999 (Auto-ID)
- ❑ Precedentes: M2M → habitualmente soluciones no interoperables
  - Cachivaches diversos conectados a IP casi por diversión
    - Máquina de bebidas conectada a ARPANET (CMU, 1982)
    - Tostadora conectada a Internet (John Romkey, 1989)
- ❑ Evolución Internet: Internet de usuarios, servicios, social y cosas
  - 6ª de las 6 webs de Bill Joy → D2D
  - Hoy, diversas interacciones Human&Things: H2H, T2T, H2T
- ❑ Variaciones: *Industrial IoT, Social IoT...*
- ❑ Enorme escala (<https://lefronic.com/internet-of-things-statistics/>)
  - 8/2019: 127 nuevos dispositivos IoT/s, 26 mil millones activos
- ❑ Aunque por debajo de las predicciones:
  - <https://spectrum.ieee.org/telecom/internet/the-internet-of-fewer-things>

# Cyber-Physical Systems vs IoT



**CPS = Sistema empotrado + Sistema físico controlado**

**IOT → CPS interconectados interaccionando entre sí**

*A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications*

Jie Lin et al. IEEE Internet of Things Journal ( Volume: 4, Issue: 5, Oct. 2017 )

# *IoT vs UbiComp*

- Distintos orígenes pero confluencia
  - *UbiComp*: (+académico) visión de alto nivel
    - Cómo vivir rodeado de computadores invisibles
    - Conectados entre sí: a Internet o de cualquier otra forma
  - IoT: visión de más bajo nivel
    - Cómo conectar objetos a Internet
- Mismas fuerzas motrices (miniaturización, redes inalámbricas...)
- Retos similares (*UbiComp*: ya vistos); IoT:
  - Escalabilidad, seguridad y privacidad, interoperabilidad,
  - Componentes con limitaciones de recursos...
- *UbiComp* enfatiza en invisibilidad y espacios inteligentes
- IoT enfatiza en “semántica”: info generada por objetos enorme
  - Cómo indexarla, procesarla, almacenarla, consultarla...

# Modelos de interacción en IoT

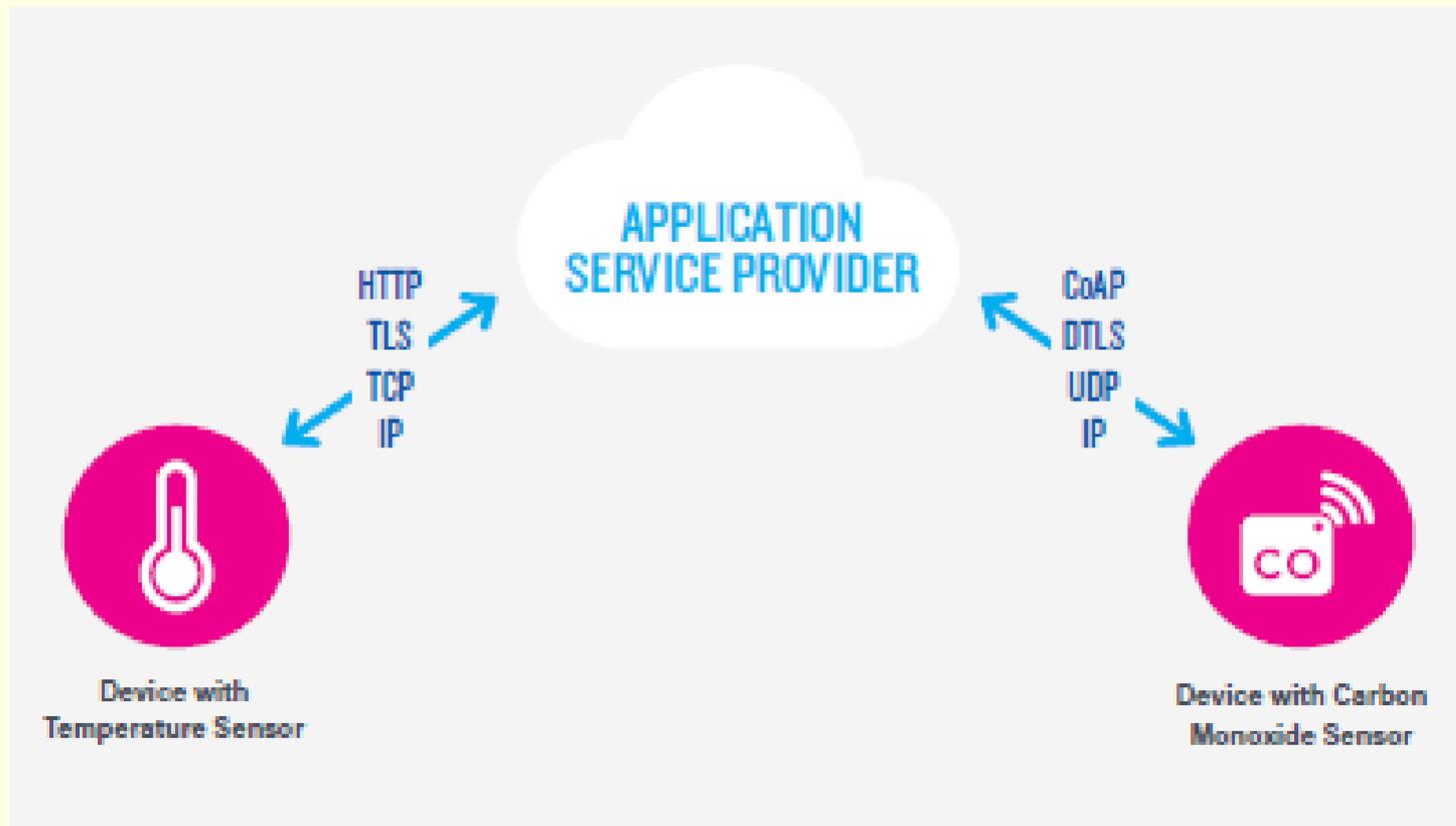
- ❑ *Device-To-Device Communication Model*
- ❑ *Device-To-Cloud Communication Model*
- ❑ *Device-To-Gateway Communication Model*
- ❑ *Back-End Data-Sharing Model*

<https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf>

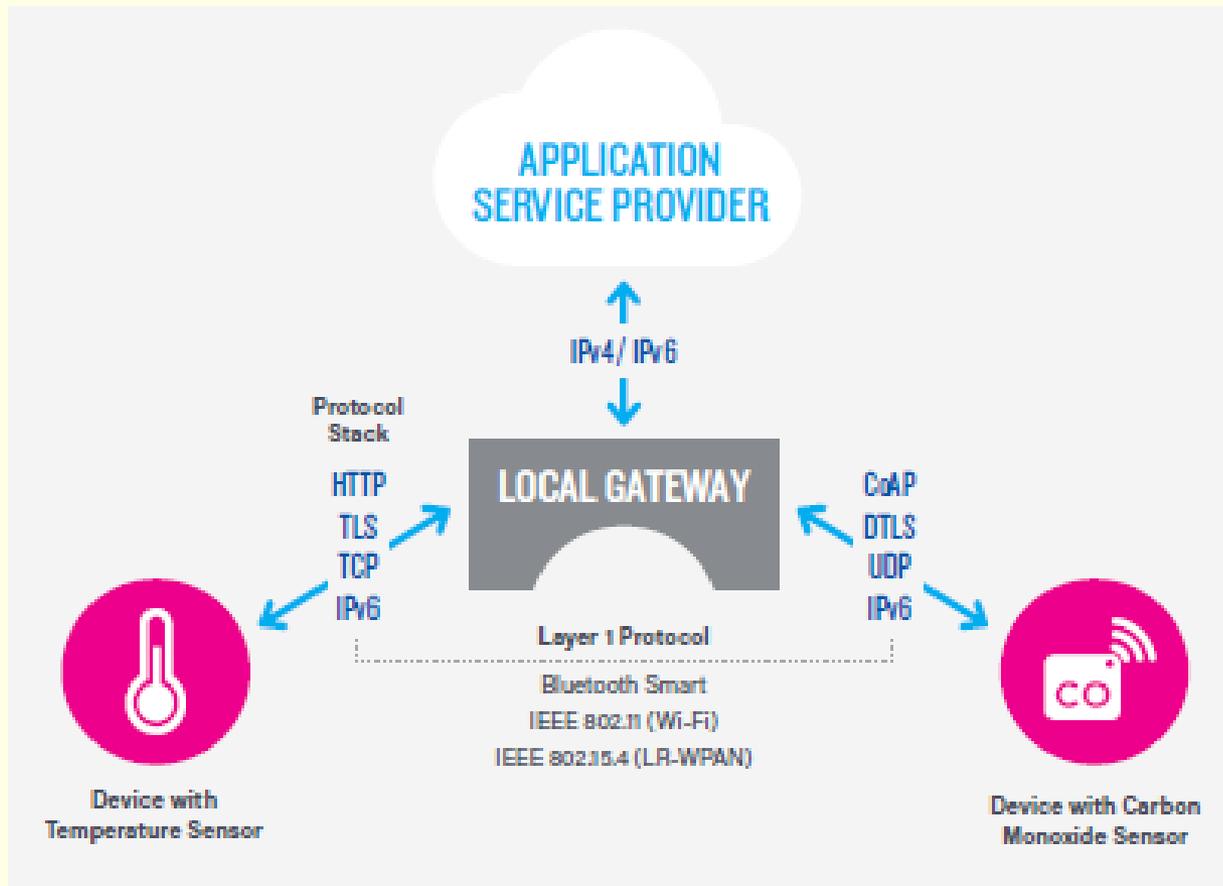
# *Device-To-Device Communication Model*



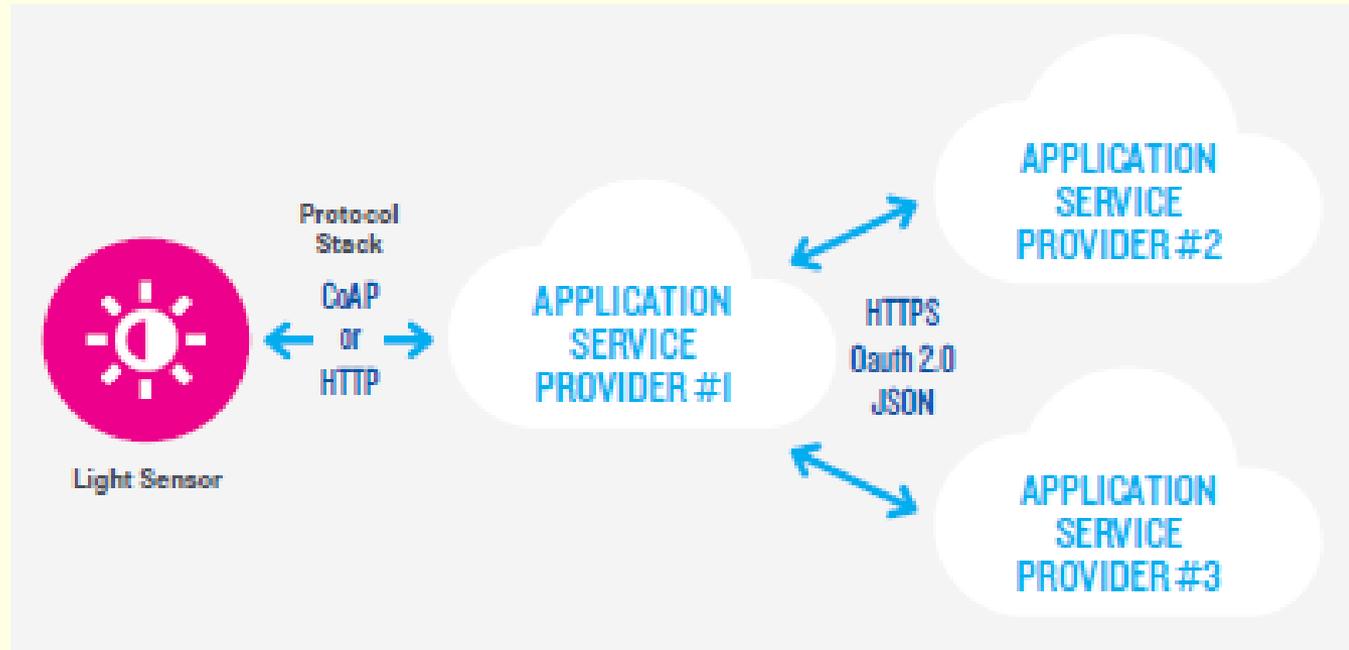
# *Device-To-Cloud Communication Model*



# Device-To-Gateway Communication Model

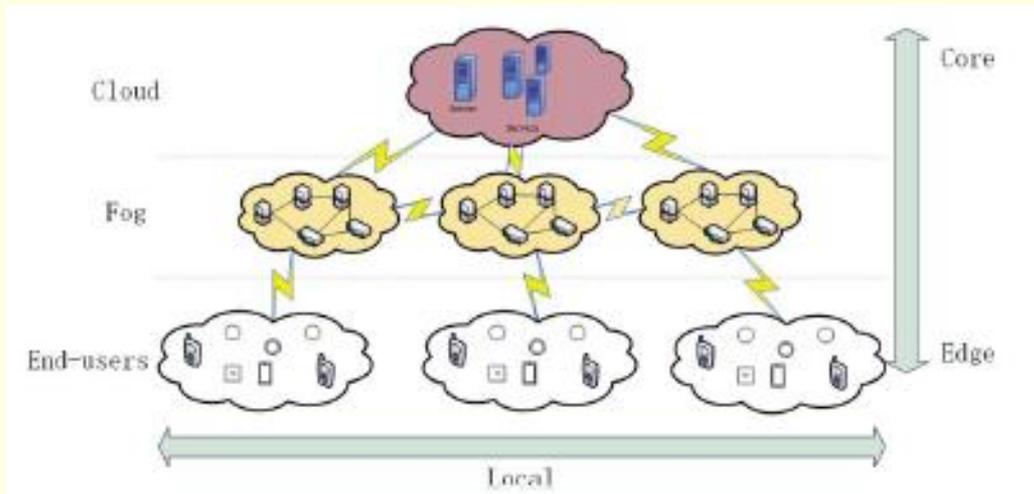


# Back-End Data-Sharing Model



# IOT *Cloud vs Fog (Edge)*

- Cantidad de Información generada por los objetos es descomunal
  - Requiere aplicación de técnicas de *Big Data*
- Solución habitual basada en *Cloud Computing* pero
  - puede requerir transferencia masiva de datos de objetos a *cloud*
- Alternativa (o, mejor dicho, complemento): *Fog/Edge Computing*
  - Acerca la computación a los objetos
  - Preprocesamiento cercano a los propios objetos
  - Mejora escalabilidad y latencia



***A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications***

# Espacios de interacción en *UbiComp*

- Un sistema *UbiComp* crea un espacio de interacción (EI):
  - dinámico, espontáneo, volátil e “inteligente”
- Usuarios y dispositivos entran y salen continuamente del mismo
  - Mi cámara digital y yo entramos en habitación de hotel
  - Vehículos entran y salen de área de control de un “semáforo”
- Puede no haber conocimiento previo entre componentes del EI
  - Interacción espontánea
- Vida de un componente en el EI:
  - ¿Cómo se incorpora un componente al EI?
  - ¿Cómo los componentes de un EI se descubren e interaccionan?
  - ¿Cómo abandona el EI?

# Interacción entre componentes en sistema ubicuo

- Nodo entra en EI:
  - Se autoconfigura y descubre, y es descubierto, por restantes
  - Si proveedor de servicios, hace conocerlos a interesados
  - Si consumidor, descubre los de otros nodos que le interesen
  - *Plug & play* de servicios/dispositivos en el sistema
    - Notificación de aparición de nuevo servicio a nodos interesados
  - Necesidad de “lenguaje” de definición y búsqueda de servicios
- Nodo abandona EI: servicio/dispositivo desaparece
  - Abandono abrupto → uso de *leases*
    - Si componente no renueva *lease* se le considera fuera del EI
- Se requiere **auto-configuración** y **descubrimiento de servicios**
  - Limitación de recursos y volatilidad pueden condicionarlos
    - *UbiComp/IoT*: Dispositivos con recursos limitados
- Ej. Jini, UPnP, Zeroconf, *Service Location Protocol* (RFC 2608)

# Ejemplo plantilla de servicio de SLP

```
service:printer://lj4050.tum.de:1020/queue1  
scopes = tum, bmw, administrator  
printer-name = lj4050  
printer-model = HP LJ4050 N  
printer-location = Room 0409  
color-supported = false  
pages-per-minute = 9  
sides-supported = one-sided, two-sided
```

Extraído de “*A Comparison Of Service Discovery Protocols And Implementation Of The Service Location Protocol*”, Christian Bettstetter y Christoph Renner

# *Autonomic Computing para UbiComp*

- S. distribuidos cada vez más complejos: necesidad de autogestión
- *Autonomic computing* (iniciativa de IBM): Sistemas *Self*-\*
  - *self-configuration|self-healing|self-optimization|self-protection*
- *Autonomic Computing* especialmente adecuado para *UbiComp*:
  - Si cada componente requiriera atención aunque fuera mínima...
- Extensión para *UbiComp*:
  - *Self-regulation*
  - *Self-learning*
  - *Self-awareness (Self-inspection/Self-decision)*
  - *Self-organization*
  - *Self-creation (Self-assembly, Self-replication)*
  - *Self-management (Self-governance)*:
  - *Self-description (Self-explanation, Self-representation)*

# Auto-configuración

- Nodo que se incorpora a EI puede necesitar configurar:
  - Dirección IP (e info asociada: máscara, *broadcast*...)
  - Nombre de máquina
- No concebible “configuración manual”
  - Acapararía atención de los usuarios
  - Necesidad de autoconfiguración
- Por limitaciones de recursos y volatilidad
  - Pueden no estar disponibles soluciones basadas en servidores
    - Ni DHCP para asignar IP
    - Ni DNS para asignar nombre

# Auto-configuración de IP

- Alternativas: con o sin DHCP
  - Puede no estar disponible por volatilidad o limitación recursos
- Con DHCP:
  - Nodo *broadcast* petición de dirección IP
  - Servidor DHCP asigna dirección IP con *lease* asociado
- Si DHCP no disponible: uso de *Link-Local Addresses*
  - IPv4: *Dynamic Configuration of IPv4 Link-Local Addresses*
    - RFC3927
    - Nodo elige su dir. IP en el rango 169.254.0.0/16
    - Usa ARP para comprobar que no está en uso
    - Si conflicto, prueba con otra
  - IPv6 da soporte directo a *Link-Local Addresses*:
    - 64 bits de red → reservado prefijo: *fe80::/10*
    - 64 bits de *host* → convencional (basado en MAC de la interfaz)

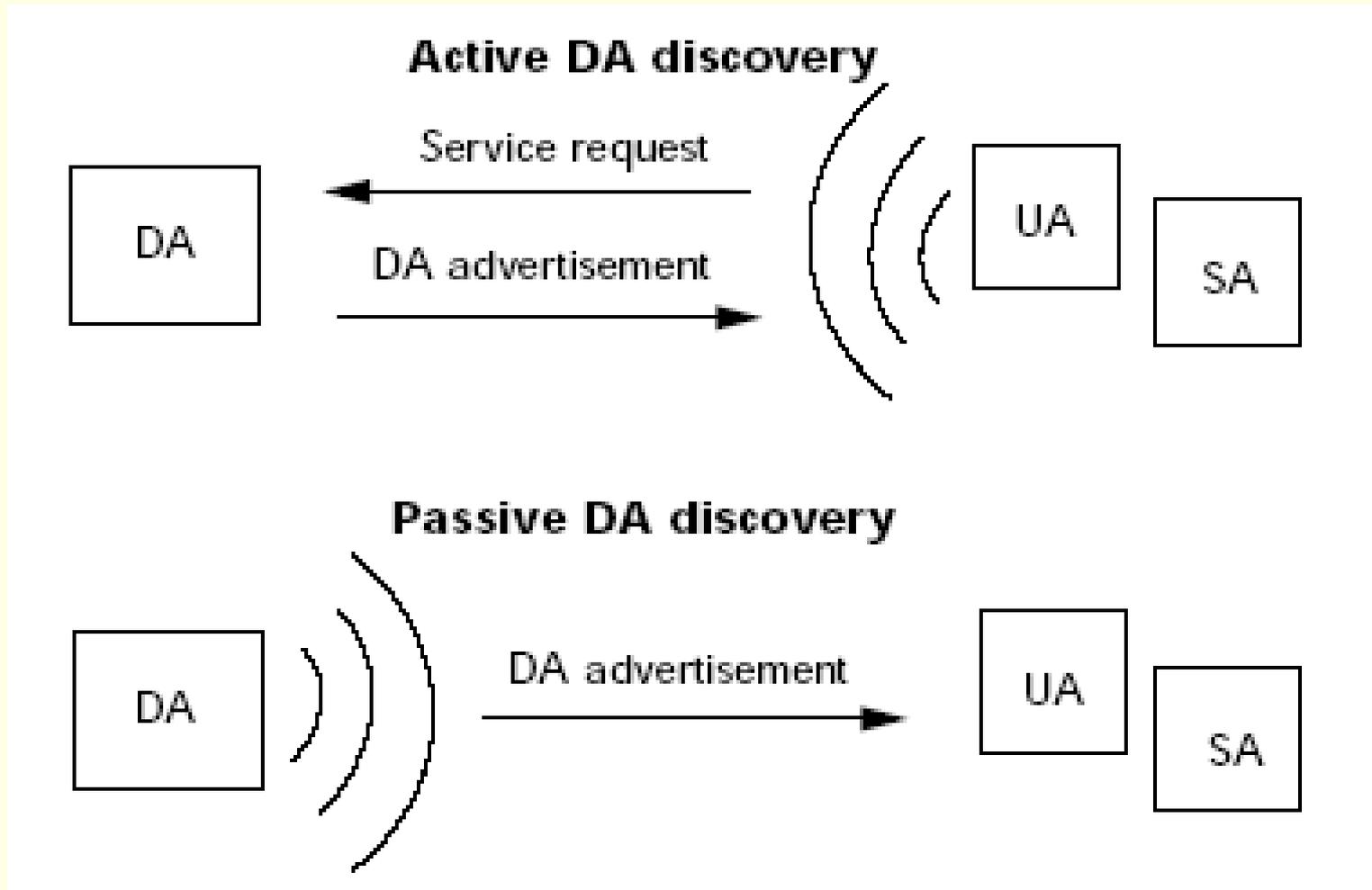
# Auto-configuración de nombre

- Asociar nombre a IP (ya sea IP por DHCP o *Link-Local*)
- Alternativas: con o sin DNS (por volatilidad o limitación recursos)
- Con DNS: Requiere habilitar “dynamic updating” (RFC 2136)
  - Equipo añade mediante *Update* el RR de tipo A: Nombre → IP
- Si DNS no disponible: *Multicast-DNS* (RFC 6762)
  - Gestiona dominio `.local.`
  - Operaciones sobre dominio `.local.` usan *multicast* dir. fija
    - 224.0.0.251 en IPv4 y ff02::fb en IPv6, con puerto UDP 5353
  - Equipo con ese nombre responde enviando su IP
  - Pueden pedir respuesta *unicast* (solo al solicitante) o *multicast*
    - Respuesta *unicast* solo molesta al interesado
    - Pero *multicast* permite a otros conocer información por adelantado
  - Equipo añade localmente RR de tipo A (Nombre → IP)
    - Se comprueba que no hay conflictos por nombres repetidos

# Servicios de descubrimiento de servicios (1/2)

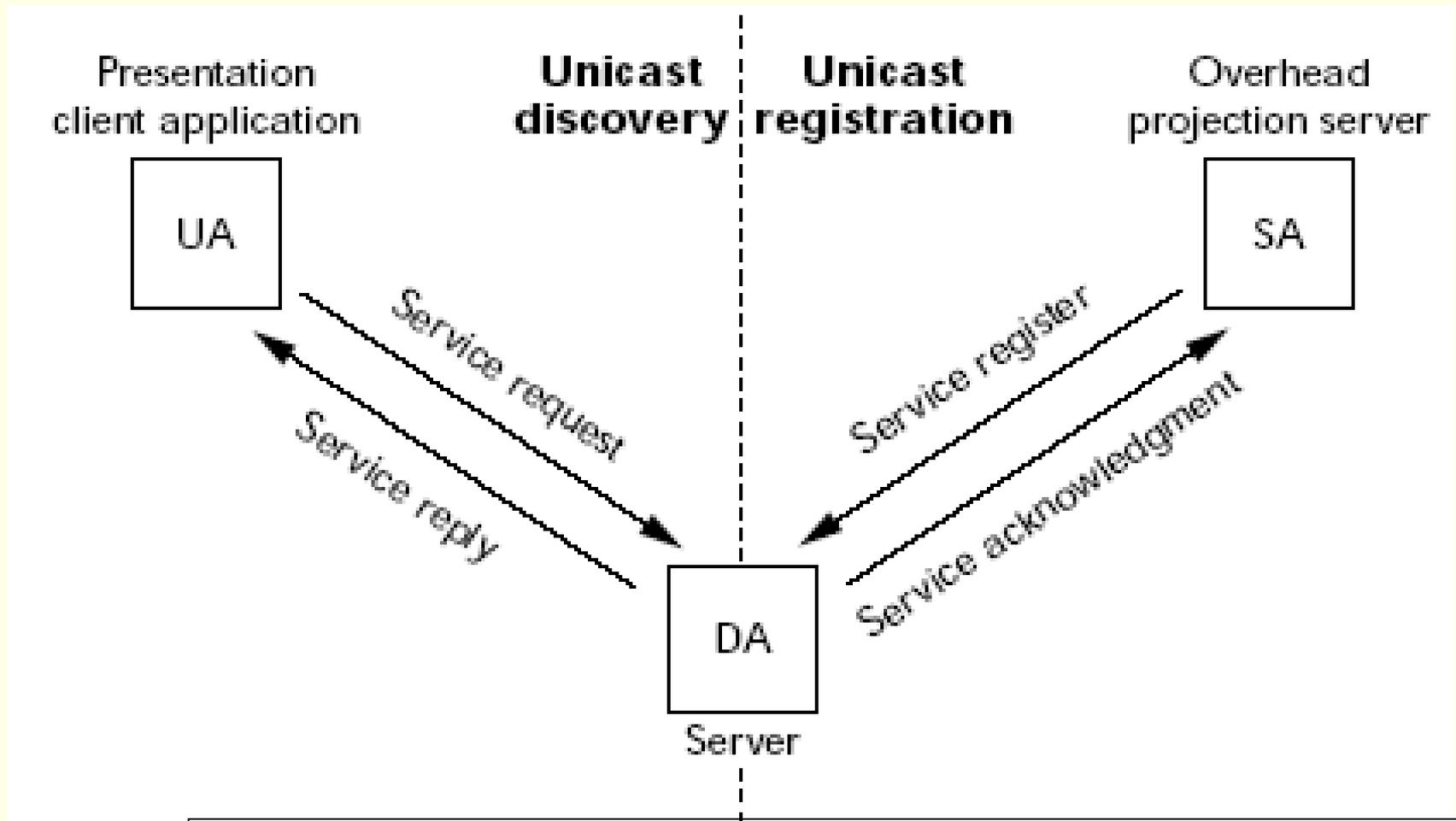
- Tres roles (usando terminología SLP):
  - cliente (UA), proveedor serv. (SA), serv. descubrimiento (DA)
- Alternativa principal: con o sin DA
  - Múltiples DA: replicación y/o info. de distintos ámbitos
- Con DAs:
  - UA y SA deben localizar DAs (posible filtro por ámbitos)
    - Localización pasiva: DA *multicast* a dirección fija
      - ▶ Facilita incorporación de nuevos DAs al sistema
    - Localización activa: UA/SA *multicast* a dirección fija
      - ▶ Descubrimiento de nuevos DAs mediante *polling* periódico
  - SA registra servicio mediante *unicast* en DAs localizados
  - UA consulta mediante *unicast* a alguno de los DAs localizados
  - UA puede pedir a DA notificación si aparece un cierto SA (PnP)

# Descubrimiento de DA en SLP



*Service Location Protocol: Automatic Discovery of IP Network Services. Erik Guttman*

# Registro y búsqueda de servicio en SLP



*Service Location Protocol: Automatic Discovery of IP Network Services. Erik Guttman*

# Servicios de descubrimiento de servicios (2/2)

- Sin DA: *pull* versus *push*
  - *Push*: SA *multicast* anuncio de servicio; UAs guardan esa info.
    - *push* descubrimiento automático nuevo SA
  - *Pull*: UA *multicast* petición; SA la recibe y responde
    - *pull* descubrimiento nuevo SA mediante *polling* periódico
- Esquema híbrido (SLP): suponiendo *pull* y descubrimiento pasivo
  - Mientras no haya ningún DA:
    - SA escucha dir. *multicast* peticiones de servicio
    - UA envía a dir. *multicast* peticiones de servicio
    - SAs/UAs escuchando dir. *multicast* posibles altas de DAs
  - Cuando aparece un DA no habiendo ninguno antes
    - SAs registran servicio en DA mediante *unicast*
    - UAs consultan DA usando *unicast*
  - Si desaparecen todos los DAs: vuelta al primer punto

# DNS-SD (RFC 6763)

- Extensión de DNS para que opere como un servicio de directorio
  - Ya sea servidor DNS con *Update* o *multicast* DNS
- Requisitos:
  - Registrar un determinado tipo de servicio
    - Especificando su IP, puerto y propiedades (clave=valor)
    - Servicio de impresión IPx, ptoY y formato=PostScript
  - Averiguar todas las instancias de un tipo de servicio
    - ¿Qué servicios de impresión están activos?
  - Obtener la información de una instancia determinada
    - IP, puerto y propiedades de esa instancia de servicio
  - Obtener qué tipos de servicios hay activos en el sistema
    - Servicios de impresión, almacenamiento, sensores...
  - Pedir ser notificado cuando aparezca un cierto tipo de servicio
    - Avísame cuando se active un servicio de impresión
  - Detectar cuando dispositivo deja EI para no usar sus servicios

# DNS-SD: Uso de *Resource Records*

- Usa SRV, PTR y TXT para especificación de servicios
- SRV en DNS convencional
  - Permite especificar varias máquinas para un **mismo** servicio
    - Para tolerancia a fallos y reparto de carga
  - No suficiente para descubrimiento de servicios
  - Se necesita definir varias instancias de un servicio
  - Extensión de SRV: *instancia.\_servicio.\_protocolo.dominio*
    - *webClientes.\_http.\_tcp.miemp.es. s1.miemp.es. 8080*
    - *webClientes.\_http.\_tcp.miemp.es. s2.miemp.es. 80*
- TXT en DNS convencional
  - Permite incluir cualquier tipo de información
  - En DNS-SD permite definir propiedades de una instancia

# Recordatorio: RR de tipo SRV

- ☐ Permite especificar qué máquinas dan un servicio en el dominio
  - y por qué puerto lo proporcionan
- ☐ Formato: *\_servicio.\_protocolo.dominio TTL clase SRV prioridad peso puerto servidor*
  - Prioridad: petición debe dirigirse a máquinas de máx. prioridad
    - Solo se envía a siguiente prioridad si no operativas
  - Peso: % de reparto de carga entre máquinas de misma prioridad

## ☐ Ejemplo de wikipedia:

```
_sip._tcp.example.com. 86400 IN SRV 10 60 5060 bigbox.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 20 5060 smallbox1.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 10 5060 smallbox2.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 10 5066 smallbox2.example.com.  
_sip._tcp.example.com. 86400 IN SRV 20 0 5060 backupbox.example.com.
```

## ☐ Ejemplo real: *dig -t SRV \_xmpp-server.\_tcp.gmail.com.*

```
_xmpp-server._tcp.gmail.com. 900 IN      SRV 20 0 5269 alt2.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN      SRV 20 0 5269 alt4.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN      SRV 20 0 5269 alt1.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN      SRV 5 0 5269 xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN      SRV 20 0 5269 alt3.xmpp-server.l.google.com.
```

## ☐ Repaso DNS:

[http://laurel.datsi.fi.upm.es/media/docencia/asignaturas/sd/sd\\_servicio\\_nombres-2019.pdf](http://laurel.datsi.fi.upm.es/media/docencia/asignaturas/sd/sd_servicio_nombres-2019.pdf)

# DNS-SD: Uso de *Resource Records*

## ☐ PTR en DNS convencional

- Permite traducciones inversas (IP→nombre)
- DNS-SD asocia un tipo de servicio con instancias del mismo
  - *.\_http.\_tcp.miemp.es. PTR webClientes.\_http.\_tcp.miemp.es.*
  - *.\_http.\_tcp.miemp.es. PTR webProveedores.\_http.\_tcp.miemp.es.*
- Meta-consulta PTR para obtener todos los tipos de servicios:
  - *\_services.\_dns-sd.\_udp.dominio*
  - Consulta PTR *\_services.\_dns-sd.\_udp.dominio*
  - Obtiene los PTR de los tipos de servicios activos en el sistema

## ☐ Extensiones DNS-SD no estándar (sí en implementaciones)

- *long-lived queries* para ser notificado si aparece nuevo servicio
  - Petición persistente
- Añade *leases* para detectar servidor ya no presente
  - O peticionario de *long-lived queries* ya no presente

# Zeroconf

- Configuración automática máquinas/servicios en red IP
- Distintas implementaciones
  - Bonjour en Apple
  - Avahi en Linux
- Selección de dirección IP
  - DHCP y, si no disponible, direcciones *Link-Local*
- Configuración nombres de máquina
  - DNS convencional y, si no disponible, *Multicast-DNS*
- Descubrimiento de servicios:
  - Usar como DA → DNS-SD
  - Ya sea con servidor DNS o *multicast-DNS*

# Ejemplos de Avahi

- ❑ 3 máquinas: *maq0*, *maq1*, *maq2* con Ubuntu 18.04
  - Mismo “segmento de red” (cableado o inalámbrico)
  - Inicialmente desconfiguradas
- ❑ Autoconfiguración IP: Mandato *avahi-autoipd*. Se encarga de:
  - Asignar LL-IPv4 (alg. asignación), LL-IPv6 (soporte directo)
  - Añadir rutas y nombre mDNS (*maq0.local,maq1.local,maq2.local*)
- ❑ Descubrimiento de servicios:
  - En *maq1* se activa servicio de sensor de humedad
  - En *maq2* se activan servicios web y sensor de temperatura
  - En *maq0* se descubren los servicios:
    1. Se determina qué tipos de servicios hay disponibles  
*PTR \_services.\_dns-sd.\_udp.local.*
    2. Se obtiene qué instancias hay de un determinado servicio  
*PTR \_sensor.\_tcp.local.*
    3. Se obtiene información de la instancia que interesa  
*SRV y TXT HumiditySensor.\_sensor.\_tcp.local.*

# Ejemplos Avahi: autoconfiguración IP

```
f Perez@maq0: ~
Archivo Editar Ver Buscar Terminal Ayuda
f Perez@maq0:~$ ip a show dev enp0s3 # interfaz no configurado
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:05:49:10 brd ff:ff:ff:ff:ff:ff
f Perez@maq0:~$ ip r # sin información de rutas
f Perez@maq0:~$ sudo avahi-autoipd -w -D enp0s3 # asigna direcciones LL
f Perez@maq0:~$ ip a show dev enp0s3 # IPs LL configuradas
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:05:49:10 brd ff:ff:ff:ff:ff:ff
    inet 169.254.2.210/16 brd 169.254.255.255 scope link enp0s3:avahi
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe05:4910/64 scope link
        valid_lft forever preferred_lft forever
f Perez@maq0:~$ ip r ; ip -6 r # información de rutas locales
default dev enp0s3 scope link metric 1002
169.254.0.0/16 dev enp0s3 proto kernel scope link src 169.254.2.210
::1 dev lo proto kernel metric 256 pref medium
fe80::/64 dev enp0s3 proto kernel metric 256 pref medium
f Perez@maq0:~$ ping -c 1 169.254.7.53 # contacta con IPv4 asignada a maq1
PING 169.254.7.53 (169.254.7.53) 56(84) bytes of data.
64 bytes from 169.254.7.53: icmp_seq=1 ttl=64 time=0.920 ms

--- 169.254.7.53 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.920/0.920/0.920/0.000 ms
f Perez@maq0:~$ ping6 -I enp0s3 -c 1 fe80::a00:27ff:fe79:6c8f # contacta con IPv6 asignada a maq1
PING fe80::a00:27ff:fe79:6c8f(fe80::a00:27ff:fe79:6c8f) from fe80::a00:27ff:fe05:4910%enp0s3 enp0s3: 56 data bytes
64 bytes from fe80::a00:27ff:fe79:6c8f%enp0s3: icmp_seq=1 ttl=64 time=0.454 ms

--- fe80::a00:27ff:fe79:6c8f ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.454/0.454/0.454/0.000 ms
f Perez@maq0:~$ dig @224.0.0.251 -p 5353 +short maq1.local # mcast comprueba dado de alta nombre maq1
169.254.7.53
f Perez@maq0:~$ avahi-resolve-host-name -4 maq1.local # lo mismo con mandato avahi
maq1.local      169.254.7.53
f Perez@maq0:~$ dig @224.0.0.251 -p 5353 +short -t AAAA maq1.local
fe80::a00:27ff:fe79:6c8f
f Perez@maq0:~$ avahi-resolve-host-name -6 maq1.local # lo mismo con mandato avahi
maq1.local      fe80::a00:27ff:fe79:6c8f
f Perez@maq0:~$ ssh maq1.local
f Perez@maq1.local's password: █
```

# Ejemplos Avahi: publicación de servicios



```
Actividades Terminal lun 09:08
Papelera
Archivo Editar Ver Buscar Terminal Ayuda
fperez@maq1: ~
fperez@maq1:~$ sudo avahi-publish-service HumiditySensor _sensor._tcp 12346 "type r
relative" & # maq1 define servicio
[1] 3611
fperez@maq1:~$ Establecido bajo el nombre «HumiditySensor»
fperez@maq1:~$

fperez@maq2: ~
Archivo Editar Ver Buscar Terminal Ayuda
fperez@maq2:~$ sudo avahi-publish-service WebClientes _http._tcp 8080 "passreq=true
" & # maq2 define servicio web
[2] 2359
fperez@maq2:~$ Establecido bajo el nombre «WebClientes»

fperez@maq2:~$ sudo avahi-publish-service TemperatureSensor _sensor._tcp 12345 "sca
le=celsius"& #maq2 define servicio
[3] 2372
fperez@maq2:~$ Establecido bajo el nombre «TemperatureSensor»
fperez@maq2:~$
```

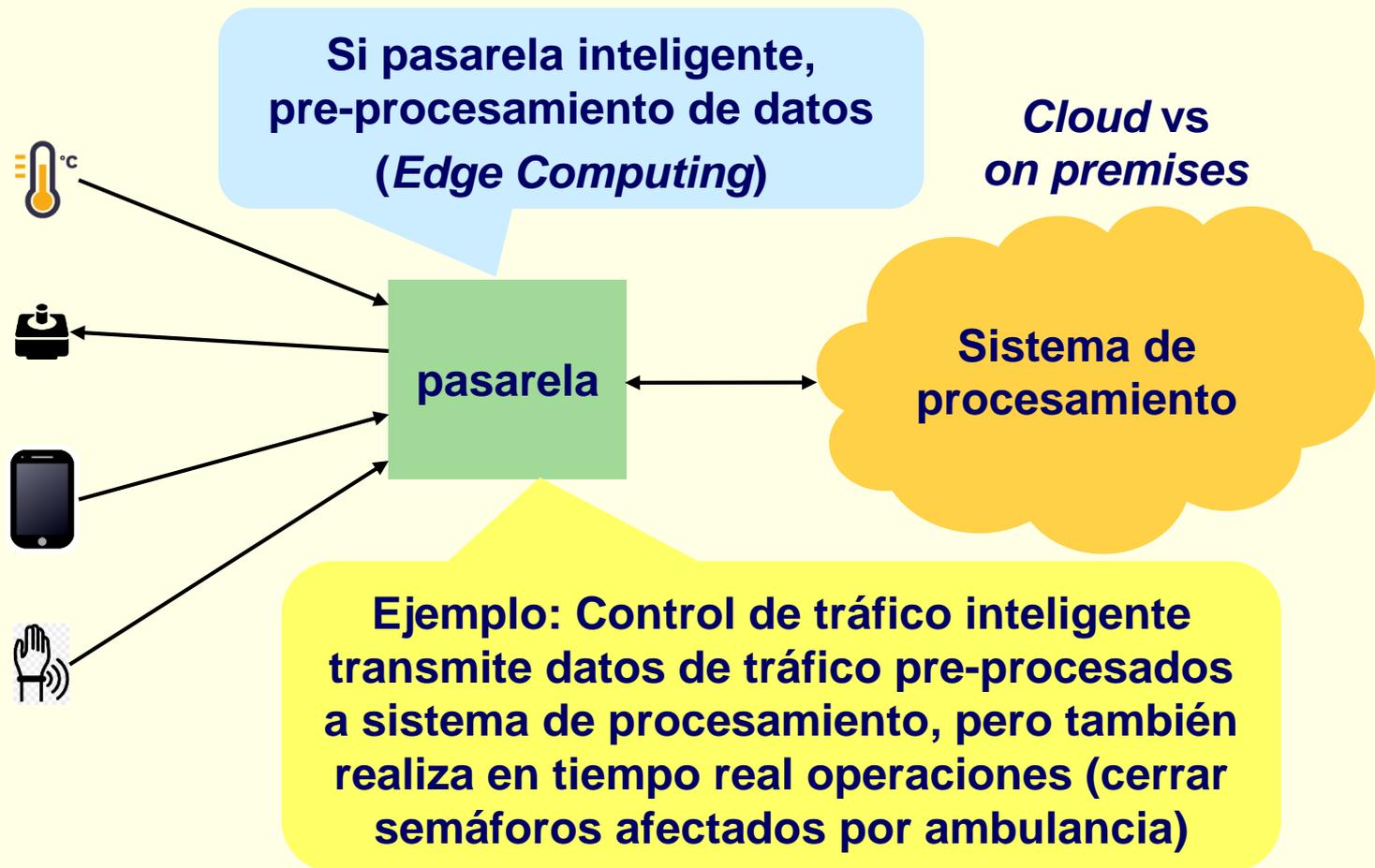
# Ejemplos Avahi: descubrimiento de servicios

```
fperez@maq0: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
fperez@maq0:~$ dig @224.0.0.251 -p 5353 +short -t ptr _services._dns-sd._udp.local. #tipos servicios  
_http._tcp.local.  
_sensor._tcp.local.  
fperez@maq0:~$ avahi-browse -tr _sensor._tcp | tail +5 # averigua información de servicios de sensor  
es  
= enp0s3 IPv6 HumiditySensor _sensor._tcp local  
  hostname = [maq1.local]  
  address = [fe80::a00:27ff:fe79:6c8f]  
  port = [12346]  
  txt = ["type relative"]  
= enp0s3 IPv4 HumiditySensor _sensor._tcp local  
  hostname = [maq1.local]  
  address = [169.254.7.53]  
  port = [12346]  
  txt = ["type relative"]  
= enp0s3 IPv6 TemperatureSensor _sensor._tcp local  
  hostname = [maq2.local]  
  address = [fe80::a00:27ff:fee9:d1bd]  
  port = [12345]  
  txt = ["scale=celsius"]  
= enp0s3 IPv4 TemperatureSensor _sensor._tcp local  
  hostname = [maq2.local]  
  address = [169.254.10.255]  
  port = [12345]  
  txt = ["scale=celsius"]  
fperez@maq0:~$ █
```

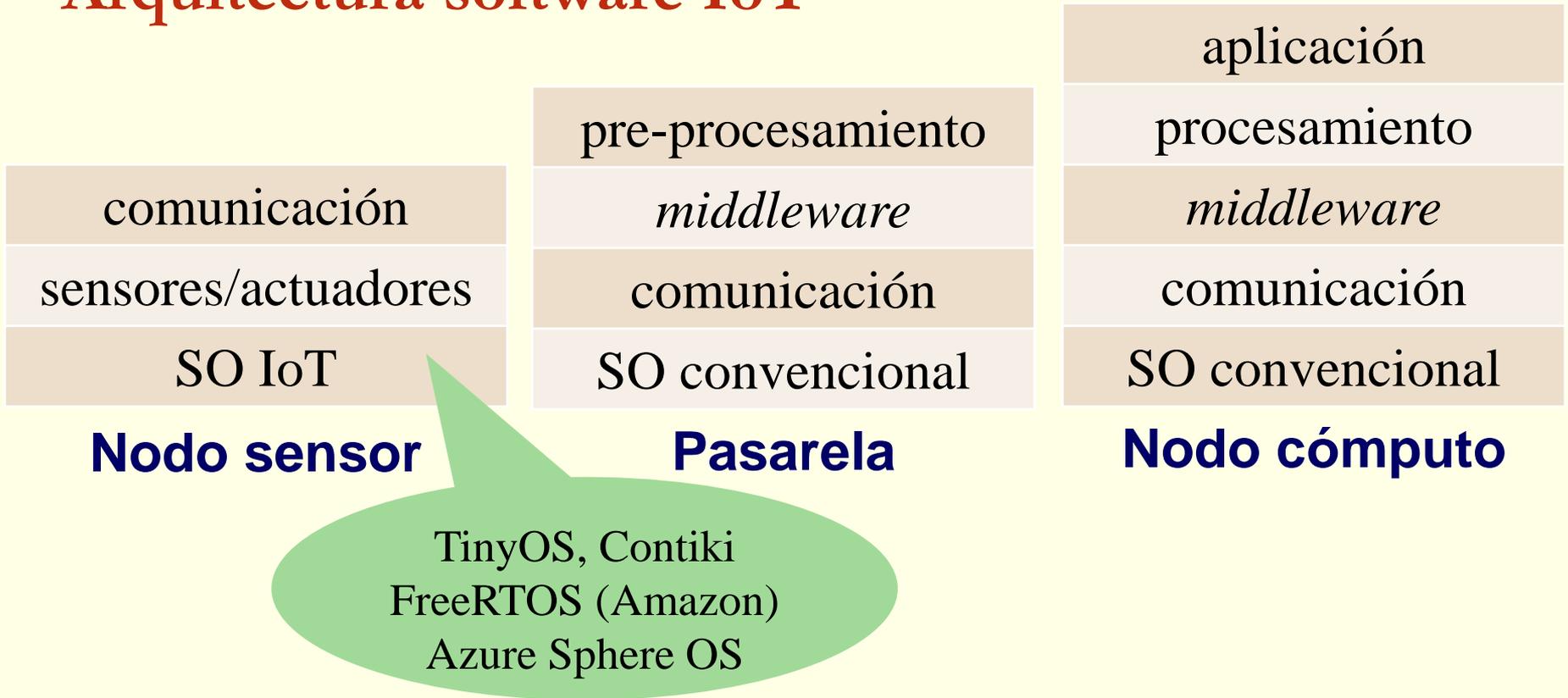
# Espacios de interacción inteligentes

- Sistema *UbiComp* emplazado en un espacio físico determinado
  - P. ej. una habitación de un hotel o las oficinas de una empresa
- Delimitación precisa de confines de un EI
  - ¡Espero que mis fotos no se impriman en habitación contigua!
- Problema del límite del EI en descubrimiento
  - ¿Cómo asegurar que asociación se restringe a límites de EI?
- Uso de un **canal físicamente restringido**
  - *Physically constrained communication channel* (Coulouris)
  - Tecnología de comunicación no traspasa límites de EI
    - IR no puede atravesar paredes de habitación de hotel
  - Codificar ID del EI en audio restringido a ese EI
    - Música de fondo muy débil en la habitación
  - Lectura ID del EI de código de barras presente en ese EI
    - Dispositivo de cliente lee código presente en habitación
  - Aproximación física entre componentes
    - Uso de comunicación de muy corto alcance o direccional

# Arquitectura de los sistemas IoT



# Arquitectura software IoT



## ▣ Revisaremos los siguientes niveles:

- Capa de sensores/actuadores
- Capa de comunicaciones
- Capa de *middleware*

# Sensores y actuadores

- Determinación de contexto comienza con lecturas de los sensores
- De muy diverso tipo (algunos en tfno. móviles o *wearables*)
  - Posición (p.e. GPS)
  - Proximidad (p.e. infrarrojo que rebota en objeto)
  - Movimiento (acelerómetro), orientación (giróscopo), podómetro
  - Magnetómetro
  - Cámara y micrófono
  - Ambientales (termómetro, barómetro, luminosidad, humedad...)
  - Químicos (polución aire y agua, *e-nose*, *e-tongue*...)
  - Biomédicos (pulso, temperatura corporal, tensión, glucosa...)
  - Identificación de personas (huella digital, iris, *Active badge*)
  - Identificación de objetos
- Actuadores: eléctricos, mecánicos, neumáticos...
  - P.e. en el hogar: cerraduras de puertas, control de luz y clima...

# Wearable Computing

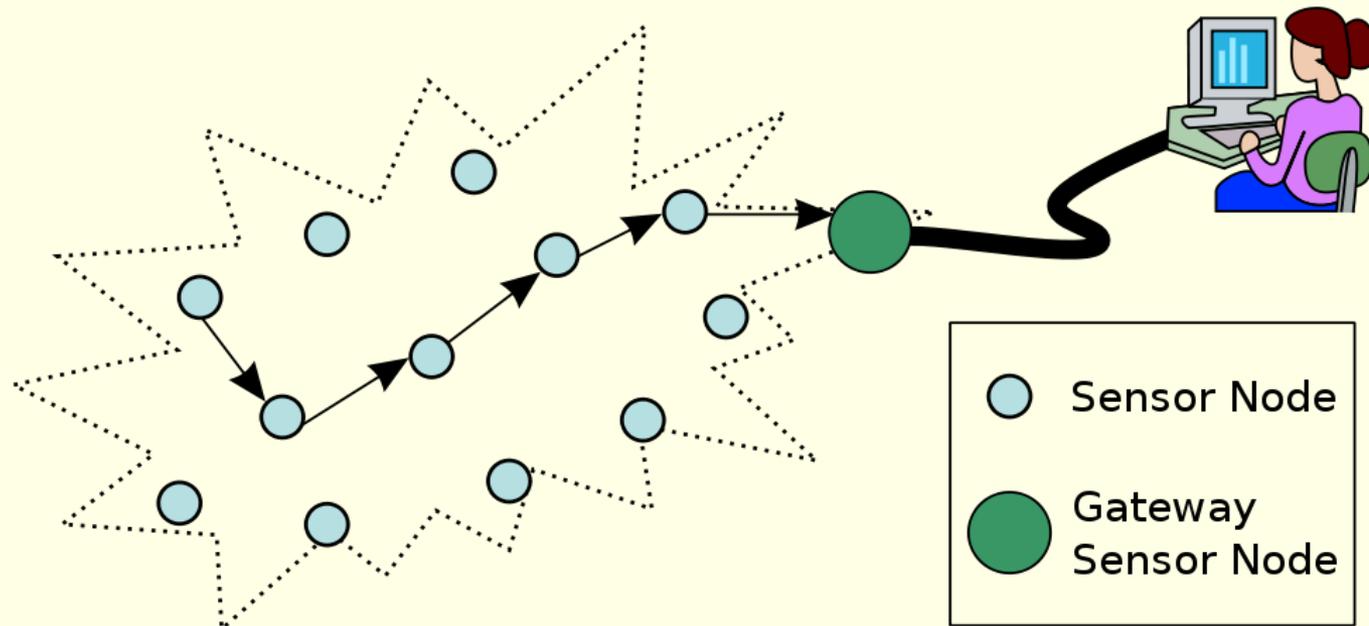
- Dispositivos que “viste” el usuario
  - En la muñeca, como gafas, en general, como parte del vestuario
- Parte integrante de *UbiComp* (e *IoT*)
  - Campo en auge
- Características deseables:
  - No deberían monopolizar la atención ( $\neq$  *smart phone*)
  - No deberían restringir el desenvolvimiento normal
  - Deberían ser controlables por el usuario
  - Debe proporcionar un *UUI*
  - Incorporan sensores (p.e. biomédicos)
  - Pueden proporcionar mecanismos de realidad aumentada
  - Debe tener capacidad para comunicarse con otros dispositivos

# Redes inalámbricas de sensores

## (*Wireless Sensor Networks, WSN*)

- Redes inalámbricas *ad hoc* de nodos sensores
  - Ideadas para funcionamiento desatendido
- Nodo sensor (*mote*): incluso tan pequeños como partículas polvo
  - Pequeño microcontrolador, sensores, batería y
  - Equipo de comunicación basado en RF con bajo consumo
  - Normalmente, corto alcance y baja potencia (p.e. Zigbee)
    - Sensores “cercaños” que envían cantidades de datos reducidas
  - Alternativa, largo alcance y baja potencia (p.e. LoRa)
    - Sensores “lejanos” que envían cantidades de datos mínimas
- Ejemplos de aplicación
  - Monitorización medioambiental de un bosque
  - Supervisión de flotas
- Propuesta: Redes de sensores RFID (RSN)

# Arquitectura de una WSN (wikipedia)

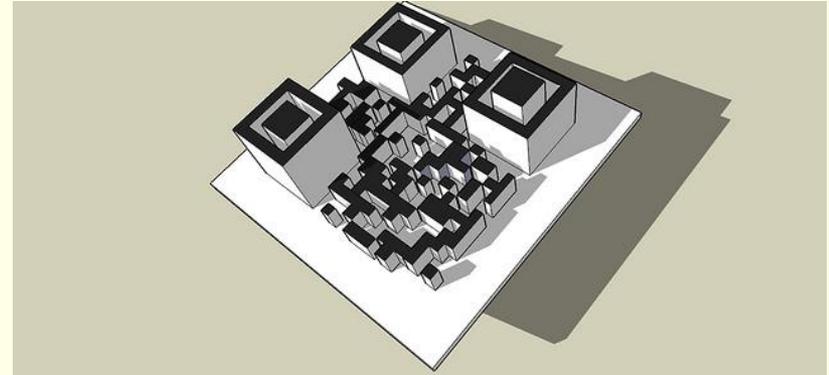
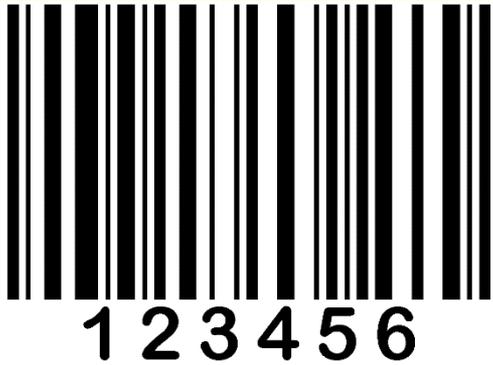


## Topologías de comunicación:

- Estrella: Nodo sensor envía directamente a pasarela
  - Problema de alcance
- Malla (véase figura): Nodos sensores y retransmisores
  - Malla plena: todo nodo sensor es retransmisor
  - Malla parcial: solo algunos nodos sensores retransmiten

# Identificación de objetos

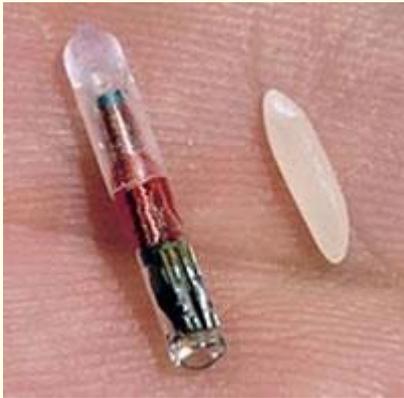
- Mecanismo para obtener ID de objeto. Alternativas:
- Código de barras
  - 1D, 2D (p.e. QR) y 3D
  - Leídos por escáner: iluminador + sensor + decodificador
- RFID (*Radio-Frequency Identification*):
  - Etiqueta: datos (ID) + antena + batería vs recolector energía
  - Lector: envía señal RF y obtiene info. enviada por etiqueta
  - Tipos:
    - **Pasiva**: sin batería; usa energía recibida del lector para responder
      - ▶ Lectura lenta (decenas de ms.), alcance corto (unos metros); WORM
      - ▶ LF (~100KHz; <10cm); HF (~10MHz; <1m) ; UHF (~1GHz; ~5m)
    - **Semi-pasiva**: usa batería; lector usa señales < energía; + caros
    - **Activa**: usa batería; etiqueta puede iniciar comunicación; + caros
      - ▶ Lectura +rápida, +alcance (100 m.); RW; +capacidad almacenamiento



# Identif. objetos: NFC (*Near Field Communication*)

- Protocolo para comunicar 2 dispositivos muy cercanos (<10cm.)
  - Adecuado para seguridad y canal físicamente restringido
  - Usa frecuencias RFID-HF pero asegurando muy corto alcance
- Dispositivo NFC tiene distintos modos de operación:
  - *Card emulation*
    - Dispositivo de pago establece conexión NFC con terminal de pago
      - ▶ Pago sin contacto (también puede basarse en RFID)
      - ▶ Monedero digital (p.e. *Apple Pay*)
  - P2P: Comunicación bidireccional entre 2 dispositivos NFC
    - Intercambian datos entre sí
  - NFC *tags*: básicamente equivalentes a RFID pasivas
    - No requieren baterías
    - Dispositivo NFC puede leerlas (e incluso escribirlas)

# *RFID y NFC tag*



<http://news.thomasnet.com/imt/2014/03/04/passive-vs-semi-passive-vs-active-tags-in-rfid>



[http://www.phonearena.com/news/10-clever-uses-for-an-Android-smartphone-with-NFC-and-NFC-tags\\_id64874](http://www.phonearena.com/news/10-clever-uses-for-an-Android-smartphone-with-NFC-and-NFC-tags_id64874)

# Comunicaciones IoT

- Por razones obvias, básicamente inalámbrica
  - Infrarrojo (IR), sonido (ultra o audible) pero
  - Nos centramos en **radiofrecuencia (RF)**
- Y con bajo consumo
- Redes IoT suelen ser LLN: *low power, lossy networks*
- Uso de IPv6 por escalabilidad
- Pero presenta problemas de adaptación al mundo IoT
  - IoT suele usar mensajes pequeños pero cabecera IPv6  $\geq 40$  bytes
  - IPv6 requiere mínimo MTU de 1280  $\gg$  MTU típica IoT (127)
  - Nodos IoT mucho tiempo dormidos
- Revisión de pila de comunicaciones IoT distinguiendo:
  - Niveles inferiores (físico PHY y enlace MAC)
  - Niveles intermedios
  - Nivel de aplicación

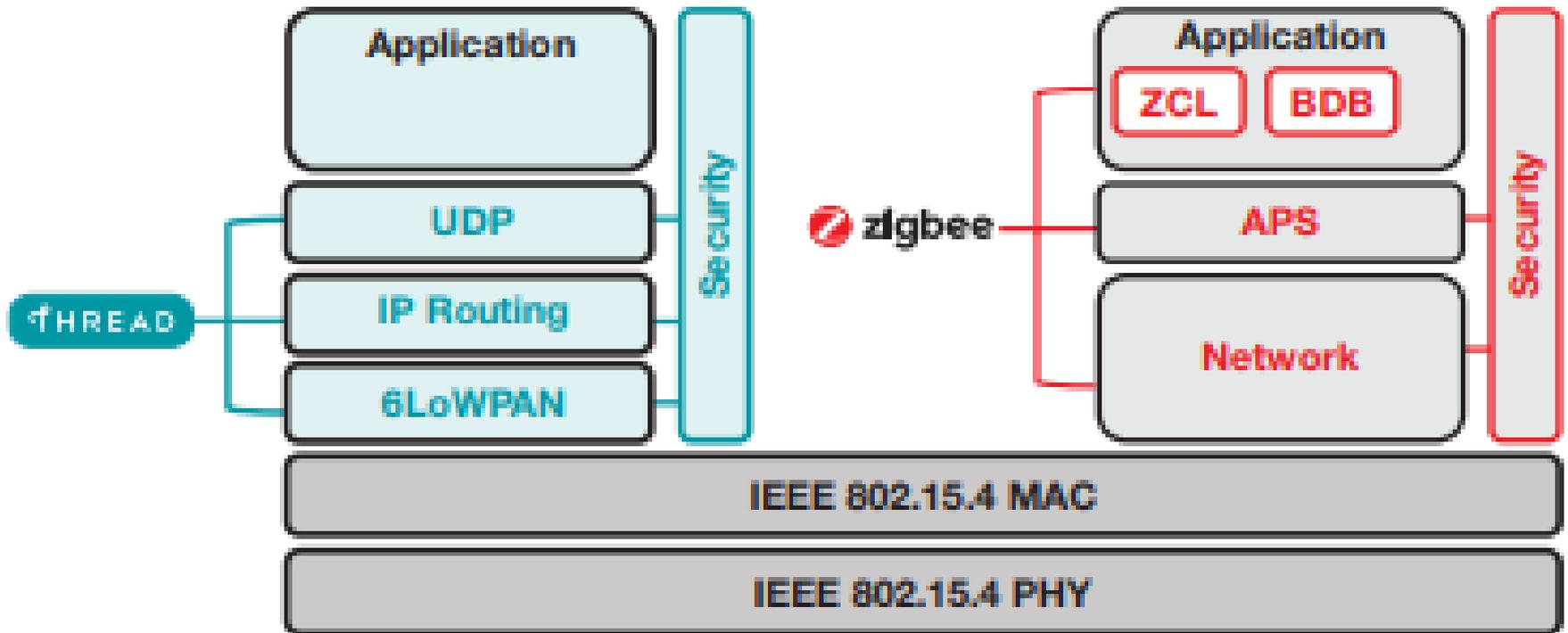
# Niveles inferiores IoT

- Tres propiedades: consumo, alcance y velocidad
  - Ideal: ↓ consumo ↑ alcance ↑ velocidad
  - Parámetros interrelacionados:
    - Mayor velocidad → mayor consumo y menor alcance
  - No podemos tener las tres (como teorema CAP)
- Si no nos importara consumo, se podría usar:
  - Corto alcance (WPAN: *Wireless Personal Area Network*):
    - p.e. Bluetooth (IEEE 802.15.1)
      - ▶ Aunque hay versiones menor consumo BLE (*Bluetooth Low Energy*)
  - Medio alcance (WAN: *Wireless Area Network*):
    - p.e. distintas versiones de Wi-Fi (IEEE 802.11)
  - Largo alcance (WWAN: *Wireless Wide Area Network*):
    - p.e. WiMAX (IEEE 802.16)
- Pero en IoT sí nos importa: ***Low-Power Wireless Area Network***
- NOTA: hay otras propiedades en las que no profundizamos
  - *Narrowband vs Wideband*, modulación (GFSK, QPSK...), expansión de la señal (FHSS, DSSS...), máximo n° nodos...

# Tipos de *Low-Power Wireless Area Network*

- ❑ LPWPAN (*Low-Power Wireless Personal Area Network*)
  - IEEE 802.15.4: alcance  $\approx 10\text{m}$ ;  $\downarrow$  velocidad  $\approx 250\text{Kbps}$
  - Varios protocolos basados en ese estándar (Zigbee, THREAD)
- ❑ LPWPAN de alta velocidad: UWB
  - Por encima de  $100\text{Mbps}$
  - Muy prometedora pero dificultades de estandarización
- ❑ LPWWAN (*Low-Power Wireless Wide Area Network*)
  - Largo alcance ( $>10\text{km}$ ) y muy bajo consumo (batería 10 años)
    - Pero velocidad  $< 1\text{Kbps}$ :
      - ▶ Ej. gasto red esperado: mensaje pequeño (16B) cada varios minutos
    - Ejemplo uso: WSN amplia, desatendida y con datos puntuales
  - Soluciones propietarias: LoRa, SigFox
- ❑ WiFi HaLow (IEEE 802.11ah): diseñada para IoT
  - Largo alcance, consumo medio, velocidad baja ( $> \text{LPWWAN}$ )
- ❑ Comunicación celular:
  - En 4G extensiones para IoT: NB-IOT y LTE-M
  - 5G soporte específico para IoT: grandes expectativas

# ZigBee vs THREAD



[https://e2e.ti.com/blogs\\_/b/process/archive/2018/05/16/thread-vs-zigbee-what-s-the-difference](https://e2e.ti.com/blogs_/b/process/archive/2018/05/16/thread-vs-zigbee-what-s-the-difference)

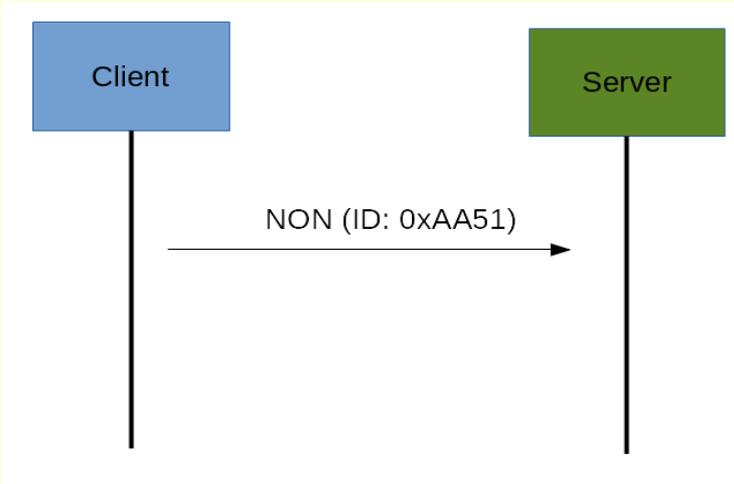
# Niveles intermedios IoT

- Dificultades de adaptar IPv6 a niveles inferiores IoT
  - Nivel de adaptación: 6LoWPAN
    - Permite transmitir IPv6 sobre 802.15.4
      - ▶ Compresión de cabeceras y fragmentación de paquetes
    - También adaptaciones a BLE, NFC, WiFi HaLow
- Nivel de transporte: Uso preferente de UDP
  - TCP demasiada sobrecarga
  - No adecuado para nodos frecuentemente dormidos
- DTLS: *Datagram Transport Layer Security*
  - Uso de DTLS (UDP) en vez TLS/SSL (TCP)
  - Incluso demasiada sobrecarga para nodos muy restringidos
- Protocolos específicos de encaminamiento:
  - RPL: *Routing Protocol for Low-Power and Lossy Networks*
- Otros protocolos específicos ya presentados:
  - DNS: *multicast DNS, DNS Service Discovery...*

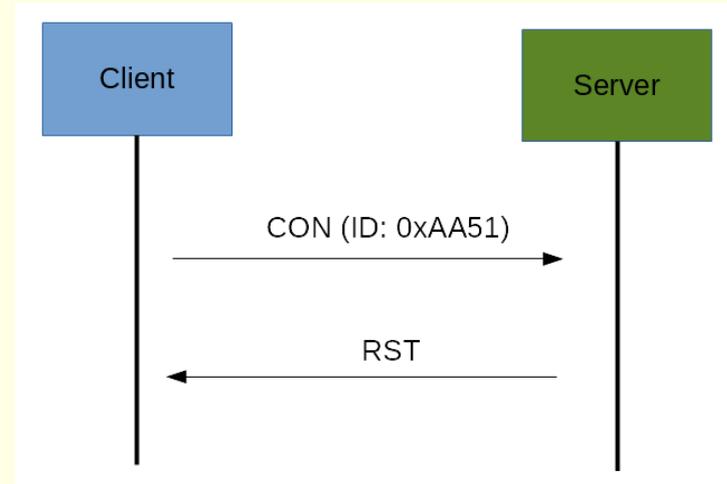
# Nivel de aplicación IoT

- HTTP no siempre adecuado para IoT
  - Basado en TCP, formato texto, cabeceras grandes, Cln/Srv
  - IoT mensajes pequeños, no siempre Cln/Srv:
    - Sensor envía unilateralmente dato a nodos interesados (PubSub)
- Niveles de aplicación alternativos: CoAP, MQTT...
- *Constrained Application Protocol* (CoAP)
  - Uso de UDP/DTLS, formato binario (EXI), cabeceras pequeñas
  - Cln/Srv compatible HTTP (GET, PUT...): soporte REST
  - Pero también de ops. asíncronas, *multicast*, petición *Observe*
- *Message Queue Telemetry Transport* (MQTT)
  - Modelo PubSub basado en temas con nodo que actúa de *broker*
  - Uso de TCP: nodos PubSub conexión persistente con *broker*
  - MQTT-SN: versión más adecuada para IoT
    - Uso de UDP, nodos dormidos pueden leer eventos a posteriori...

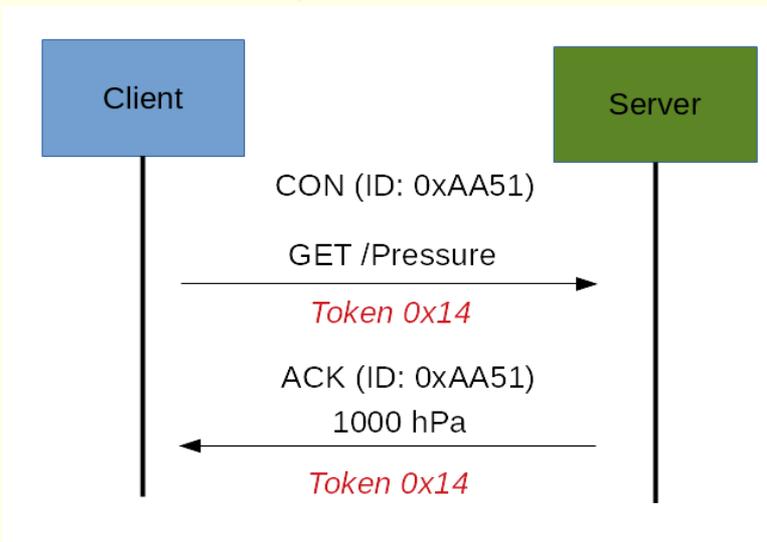
# Mensajes CoAP



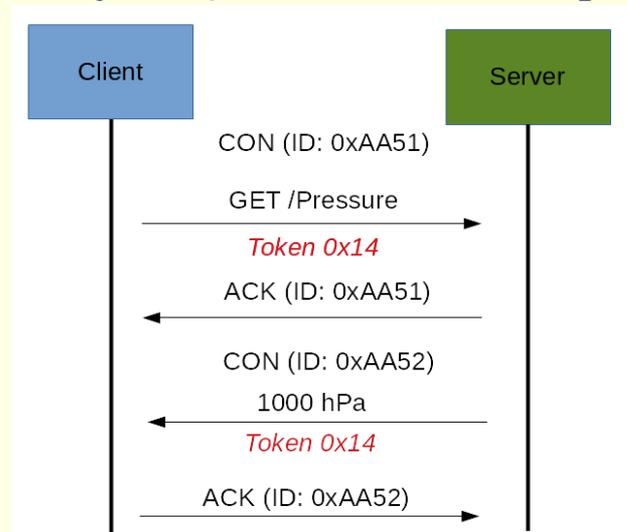
Mensaje no *confirmable*



Mensaje *confirmable* error recepción



Mensaje *confirmable* síncrono



Mensaje *confirmable* asíncrono

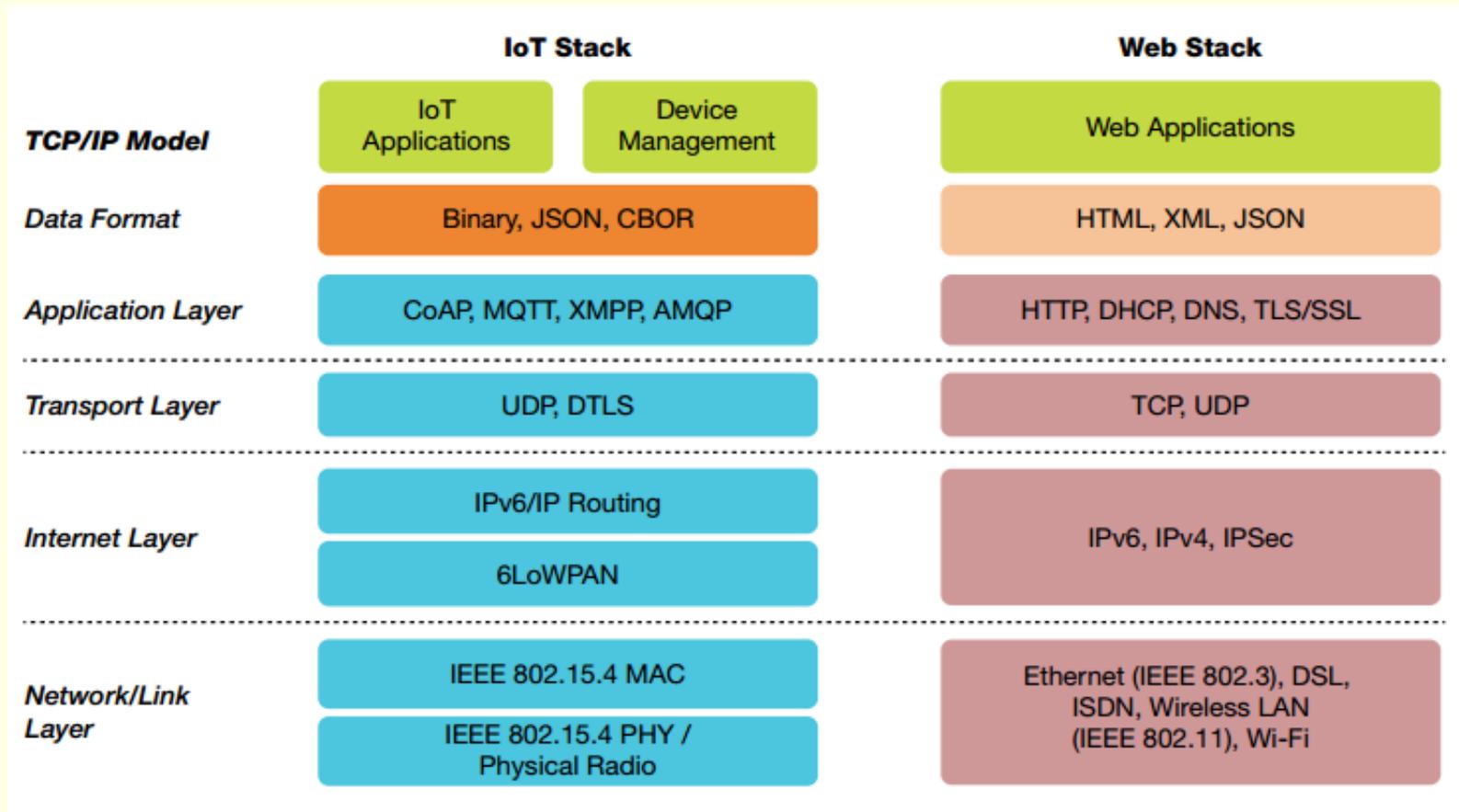
<https://dzone.com/articles/coap-protocol-step-by-step-guide>

# CoAP vs HTTP

| Feature                   | CoAP  | HTTP  |
|---------------------------|---|---|
| Protocol                  | It uses UDP.  | It uses TCP.  |
| Network layer             | It uses IPv6 along with 6LoWPAN.  | It uses IP layer.   |
| Multicast support         | It supports.  | It does not support.  |
| Architecture model        | CoAP uses both client-Server & Publish-Subscribe models.                  | HTTP uses client and server architecture.                               |
| Synchronous communication | CoAP does not need this.  | HTTP needs this.  |
| Overhead                  | Less overhead and it is simple.   | More overhead compare to CoAP and it is complex.                        |
| Application               | Designed for resource constrained networking devices such as WSN/IoT/M2M. | Designed for internet devices where there is no issue of any resources. |

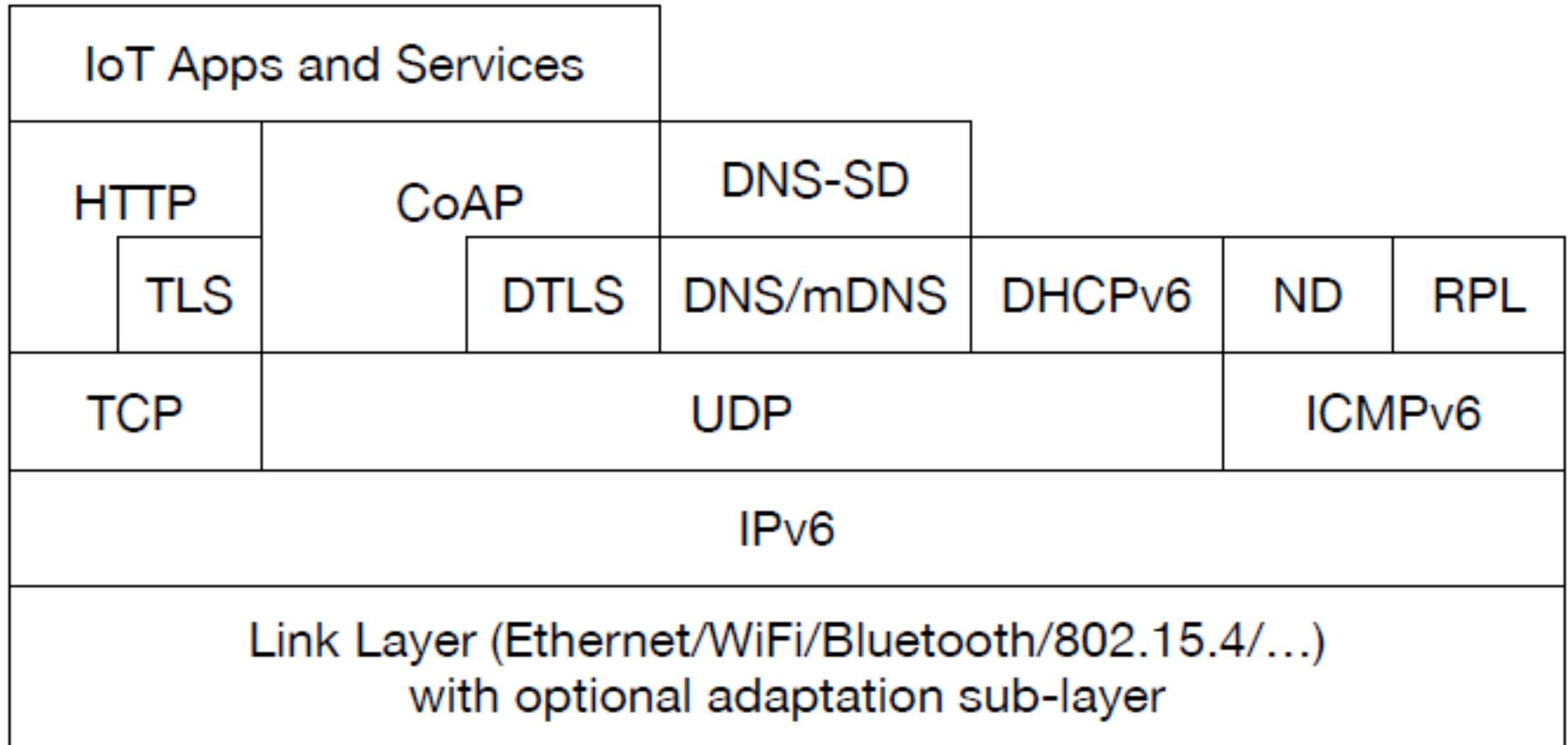
<http://www.rfwireless-world.com/Terminology/Difference-between-CoAP-and-HTTP.html>

# Pila de red IoT



<https://www.linkedin.com/pulse/emerging-open-standard-protocol-stack-iot-aniruddha-chakrabarti>

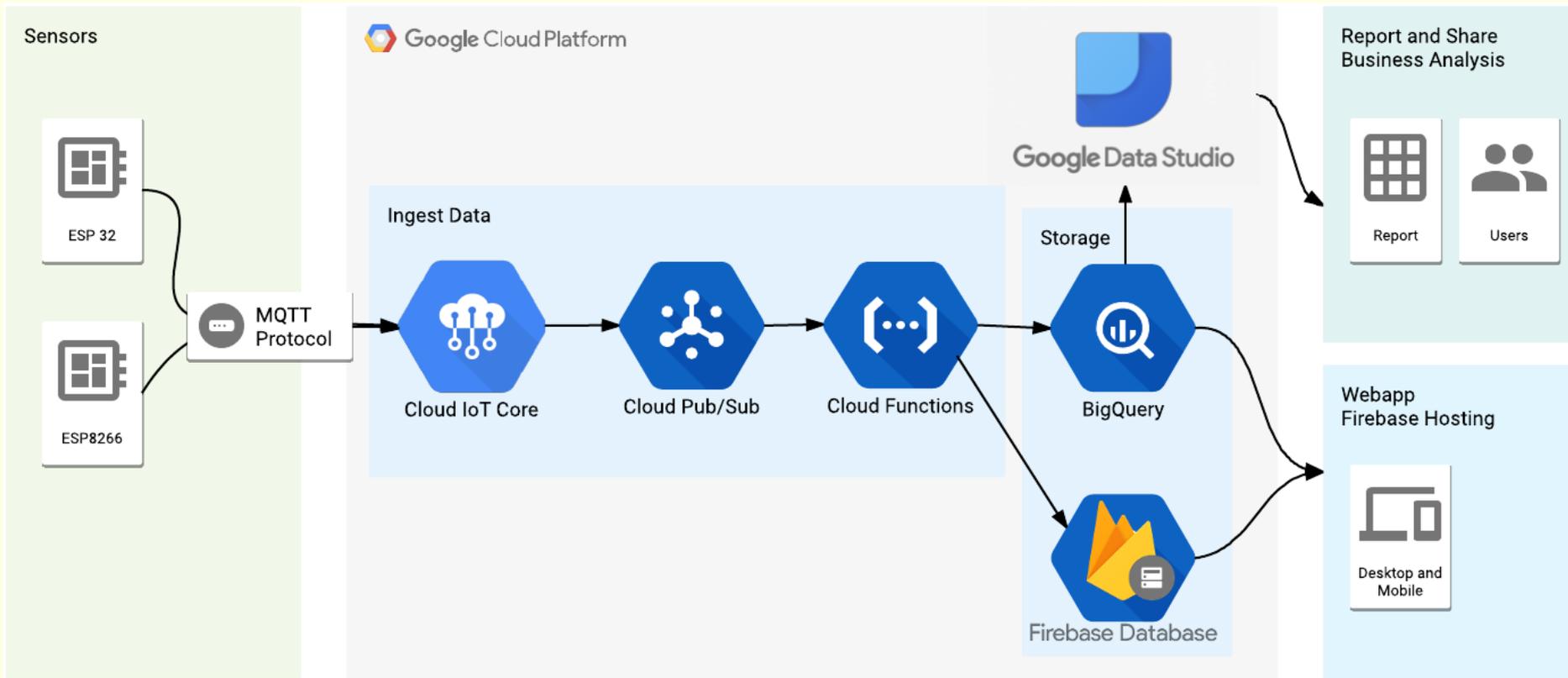
# Pila de red IoT



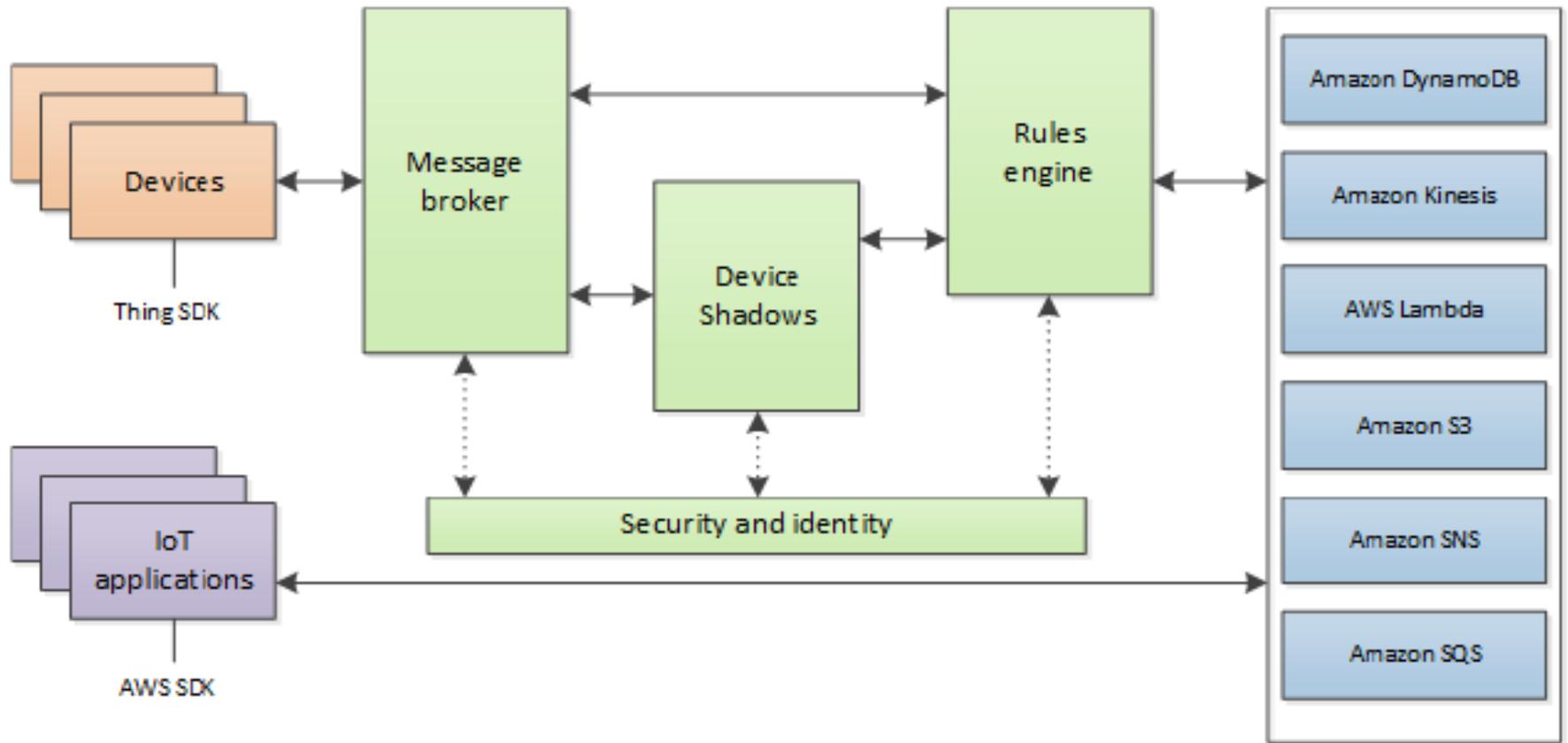
# *Middleware*

- ❑ Abstrae detalles de las *cosas* a las aplicaciones
  - Software puente entre dispositivos y aplicaciones
- ❑ Ofrece comunicación e interoperabilidad entre componentes
- ❑ Facilita gestión escalable de dispositivos
- ❑ Proporciona soporte para Big Data
  - Almacenamiento, procesamiento y análisis de datos de sensores
- ❑ Provee seguridad y privacidad
- ❑ Soluciones generalmente basadas en eventos y/o servicios
- ❑ Habitualmente en un entorno *Cloud*
- ❑ Grandes empresas se han posicionado con soluciones *Cloud IoT*
  - Google Cloud IoT
  - AWS IoT
  - Azure IoT

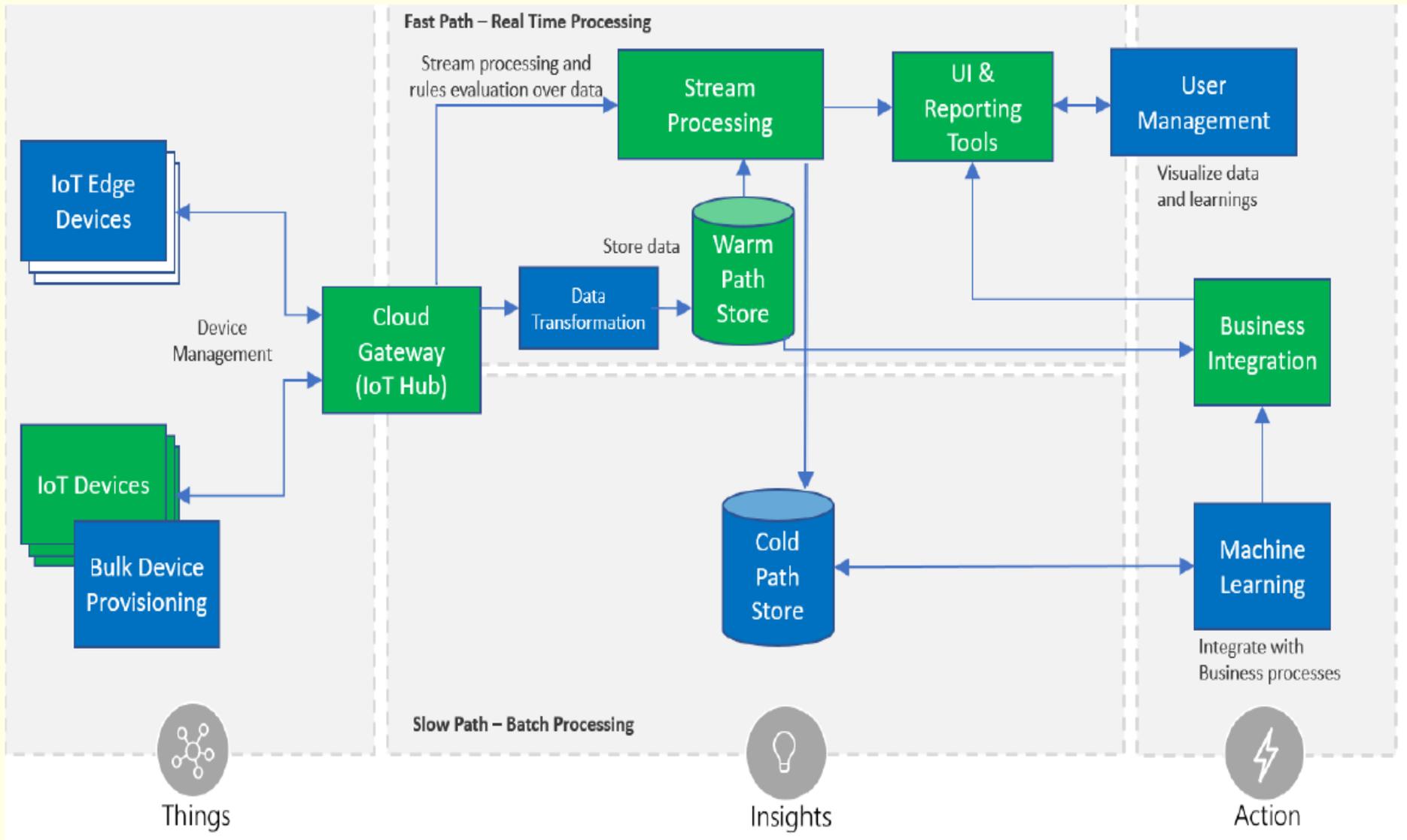
# Arquitectura Google *Cloud IoT*



# Arquitectura *AWS IoT*



# Arquitectura *Azure IoT* (lambda)



# Contexto

- Sistema CU integrado en mundo físico
  - Tiene que conocer/sentir el mundo exterior → *Sensores*
- Contexto (mundo externo + usuario + dispositivo)
  - Entorno de computación
    - HW disponible, niveles de conectividad y energía actuales
  - Entorno del usuario
    - Ubicación, gente próxima, situación social
  - Entorno físico
    - Climatología, iluminación, nivel de ruido
  - ¿Dónde, quién, cuándo, qué?
- Del contexto se infiere una determinada situación de un usuario
  - Es difícil conocer realidad y eliminar ambigüedades

# *Context-aware computing*

- Aplicación cuyo comportamiento depende del contexto
  - Se **adaptan** al contexto
    - Incluye adaptación a limitaciones del dispositivo
- El contexto puede influir en aspectos tales como:
  - Manera de presentar información y servicios
  - Ejecución automática de servicios
  - Etiquetado de datos con info. contexto para posterior uso
- Objetivo final de aplicación *context-aware*
  - Hacer lo querría el usuario en el contexto actual
    - Contexto + intenciones del usuario → acción
  - Ambigüedades en contexto y en expresión de intenciones
    - Prudencia con la pro-actividad

# Sistemas de localización

- Ubicación (dispositivo | usuario): elemento principal del contexto
- Características de un sistema de localización
  - Representación de información de ubicación
    - Física (latitud/longitud; GPS) o simbólica (“oficina 7”; *active badge*)
    - Absoluta o relativa (respecto a un punto conocido)
  - Obtenida por objeto (GPS) o por infraestructura
    - Objeto: mejor privacidad pero más cómputo
    - Infraestructura: Para localización, objeto emite señal
      - ▶ Periódica (*active badge*): consumo continuo de energía
      - ▶ Como respuesta (RFID, *ActiveBat*)
  - De interior (*ActiveBat*) o de exterior (GPS)
  - Resolución y grado de exactitud
  - Frecuencia de actualización y latencia
  - Escala (mundial, metropolitana, edificio...)
  - Privacidad y seguridad
  - Carácter intrusivo
  - Coste
  - Técnica de localización y tipo de señal utilizada

# Técnicas de localización

- ☐ Estimar posición de un objeto móvil (OM)
- ☐ Conocidas las posiciones de objetos de referencia (OR)
- ☐ Algunas técnicas de localización:
  - Proximidad
  - Trilateración (ToA/ToF, RTT/TWR)
  - Multilateración (TDoA)
  - Triangulación (AoA)
  - Mapas de huella de intensidad de señal (*Fingerprinting*)
  - Navegación por estima (*dead reckoning*)
- ☐ Pueden combinarse

# Técnicas de localización

- **Proximidad:** cercanía a OR. Posición estimada OM = posición OR
- **Trilateración (a):** Basada en distancia a ORs
  - Conocida distancia a 4 ORs se determina posición de OM
    - Punto de intersección de 4 esferas
  - ¿Cómo medir distancia? Alternativas
  - *Time of Arrival (ToA) / Time of Flight (ToF)*
    - ORs emiten su posición y tiempo actual
    - Tiempo recepción + Velocidad transmisión → distancia OR a OM
    - Requiere sincronización de relojes entre ORs y OM
      - ▶ Muy precisa para señales de radiofrecuencia
      - ▶ Coste aceptable para ORs pero no para OM
  - *Round Trip Time (RTT) / Two Way Ranging (TWR)*
    - Señal con viaje de ida y vuelta
    - Elimina necesidad de sincronización
  - *Received Signal Strength Indicator (RSSI)*
    - Atenuación de la señal recibida depende de distancia
    - Receptor conoce potencia de emisión

# Técnicas de localización

## ■ *Multilateración (TDoA: Time Difference of Arrival) (b)*

- Diferencia en t. llegada señal emitida por OM a ORs
- 4 ORs permiten determinar posición del objeto
  - Punto de intersección de 4 hiperboloides
- Requiere sincronización entre ORs pero no de OM
- Opción inversa: ORs envían simultáneamente señal
  - OM calcula *TDoA*

## ■ *Triangulación (c):*

- Dif. en ángulo llegada señal emitida por objeto a ORs
  - Requiere múltiples antenas direccionales
- 3 ORs permiten determinar posición del objeto
- Requiere sincronización entre ORs pero no de OM

# Técnicas de localización

## ■ Mapas de huella de intensidad de señal (*Fingerprinting*)

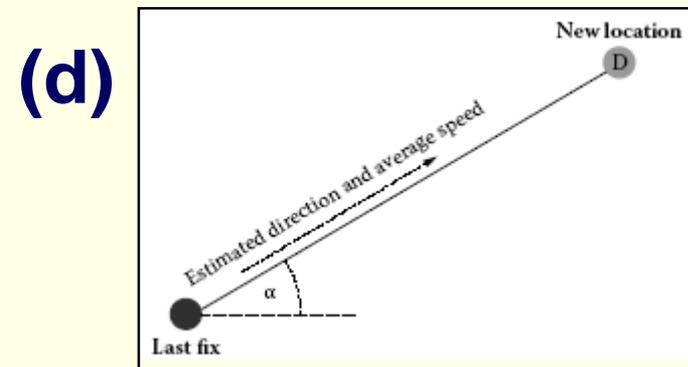
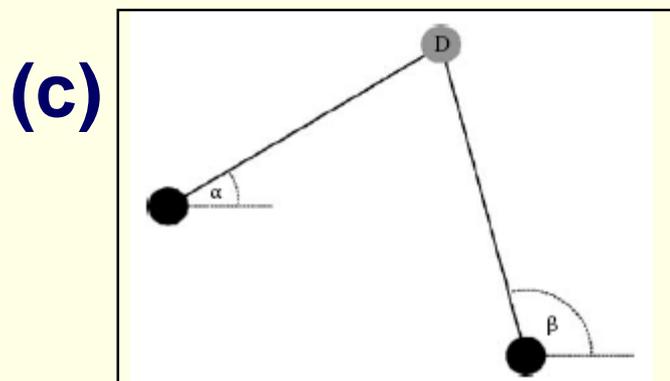
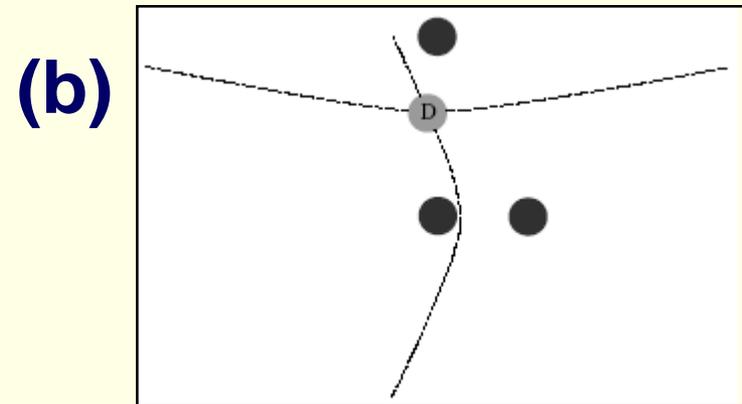
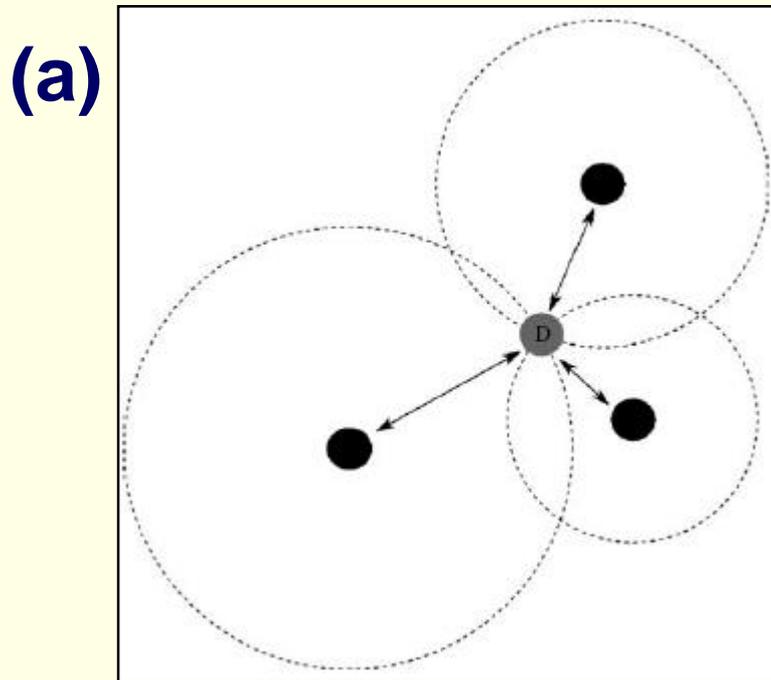
- Uso en sist. con infraestructura de emisión de señal fija
  - Intensidad recibida RSSI en un punto debería mantenerse
- Fase de calibración/entrenamiento obtiene mapa en varios ptos.
- Fase de operación: encaje en mapa de medida obtenida
  - Algoritmos de reconocimiento de patrones (k-NN)

## ■ *Inertial Navigation Systems (dead reckoning) (d)*

- Requiere técnica previa para obtener posición inicial
- Nueva posición  $\approx$  Posic. previa, dirección y velocidad estimada

## ■ Análisis de escena visual

# Estrategias para obtener la ubicación (extraído de “*Ubiquitous Computing Fundamentals*”)



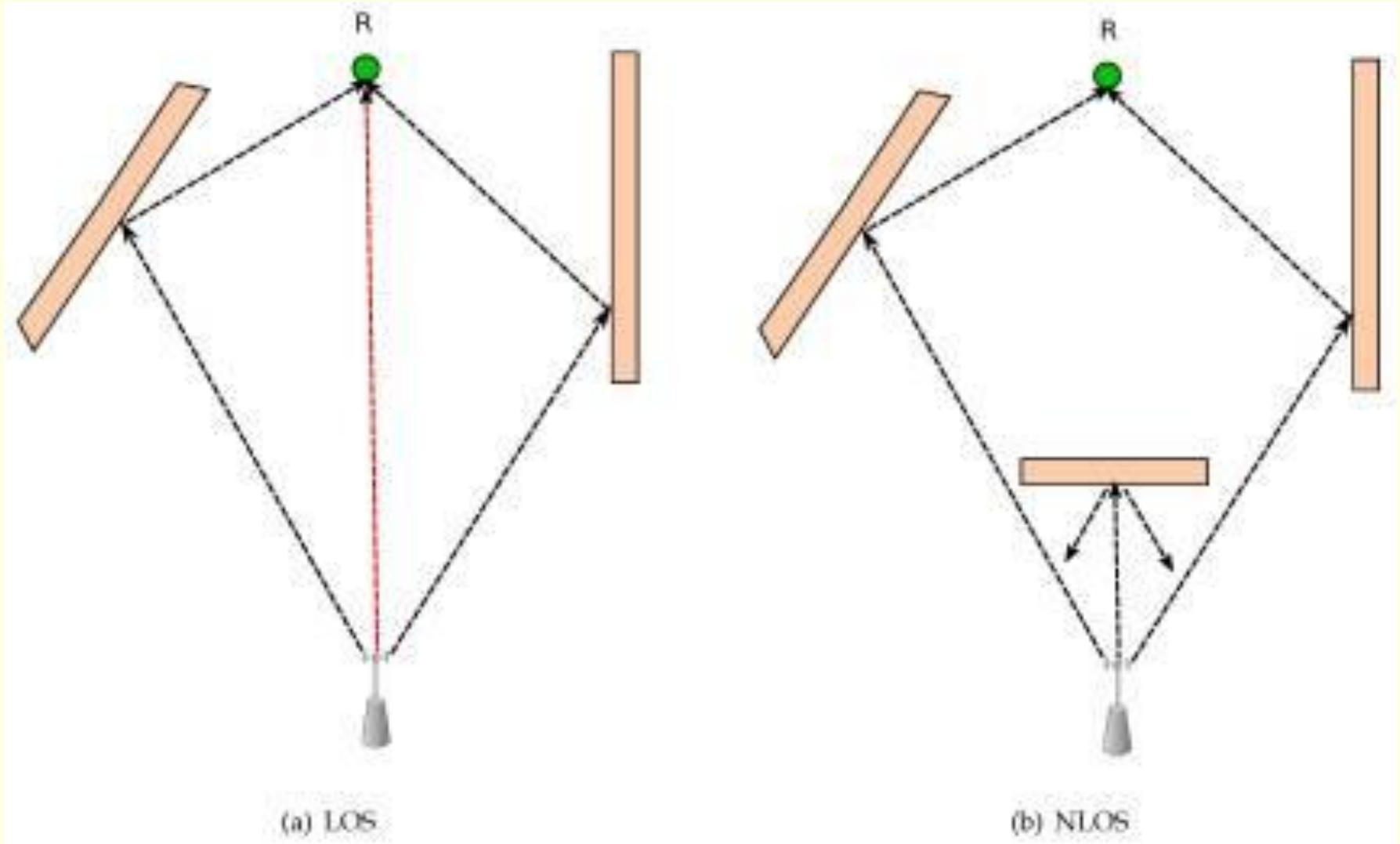
# Localización de exterior vs de interior (RTLS)

- ☐ Solución dominante para localización de exterior
  - *Global Navigation Satellite Systems* (GNSS), como GPS
  - Pero no funcionan adecuadamente en interior
- ☐ No hay una solución dominante para interior por el momento
- ☐ Usuario debe seleccionar esquema acorde a sus necesidades
  - Localización de equipo de bomberos requiere baja latencia
    - Aunque suponga un coste apreciable
- ☐ Dificultades de localización en interior:
  - Propagación multi-camino por obstáculos
  - No siempre línea de visión (*NLoS: Non-Line-of-Sight*)
  - Obstáculos que pueden cambiar de posición
  - Necesidad de precisión
- ☐ Facilidades de localización en interior:
  - Espacio pequeño donde puede instalare infraestructura
  - Ambiente climatológico estable
  - Movimientos lentos

# *Global Positioning System (GPS)*

- Sistema posicionamiento GNSS con las siguientes características:
  - Pos. física mundial, trilateración (ToA), OM calcula posición
- ORs: 24 satélites geo-síncronos orbitando; OM: receptor GPS
- ORs con relojes atómicos (sincronizados) pero OM no
- OR envía su posición y tiempo
- En principio, bastarían 3 ORs para receptor determine posición:
  - Obtiene 2 puntos pero sólo uno de ellos será razonable
  - Necesario 4º OR para corregir falta sincronización reloj OM
    - Usa TDoA para sincronizar reloj
- Precisión limitada por problemas transmisión señal ( $\approx 10$  m.)
- Alternativa: *Differential GPS* (precisión  $\approx 10$  cm.)
  - Estaciones terrestres en posición conocida calculan desviación
  - Retransmiten desviación a OMs cercanos para reajuste
- Arranque lento OM: info completa ORs y cálculos complejos
  - *Assisted-GPS*: info inicial obtenida externamente (ISP o CNP)

# Visibilidad directa y multi-camino



<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883398/>

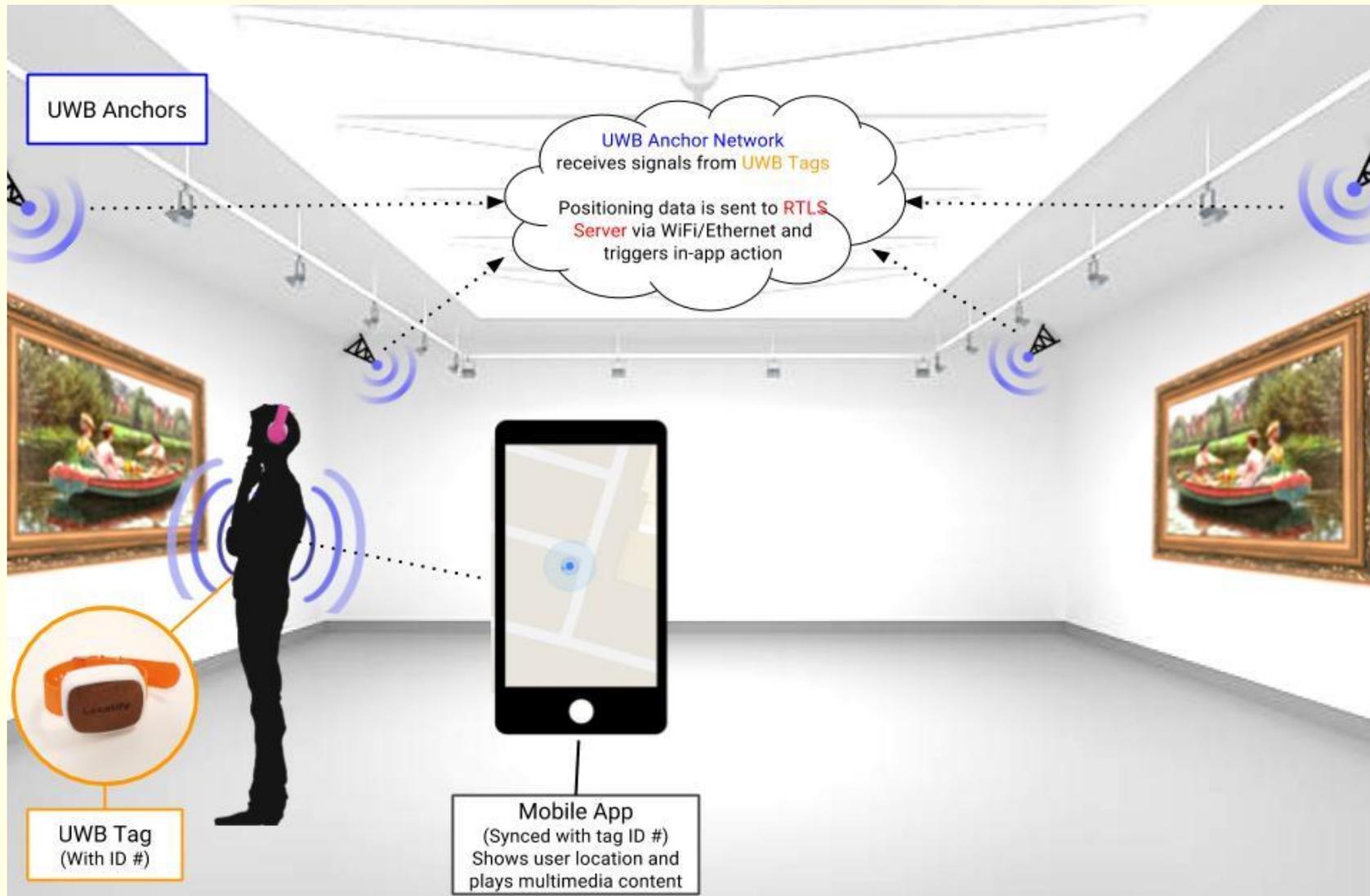
# Algunos esquemas de localización en interior

- ☐ Wi-fi:
  - Características de la señal dificultan ToA, TDoA y AoA
  - Proximidad a AP o
  - *Fingerprinting* con RSSI de señales RF recibidas desde APs
- ☐ UWB:
  - Características de la señal permiten ToA, TDoA y AoA
    - Baja interferencia, consumo, problemas multi-camino
  - Solución destacada
- ☐ Celular 5G: mejora significativa en soporte para localización
  - Mucha mayor densidad de “estaciones base”
  - Solución prometedora
- ☐ Etiquetas RFID:
  - Detección de proximidad
  - OM incluye lector
  - Etiquetas distribuidas por recinto

# Algunos esquemas de localización en interior

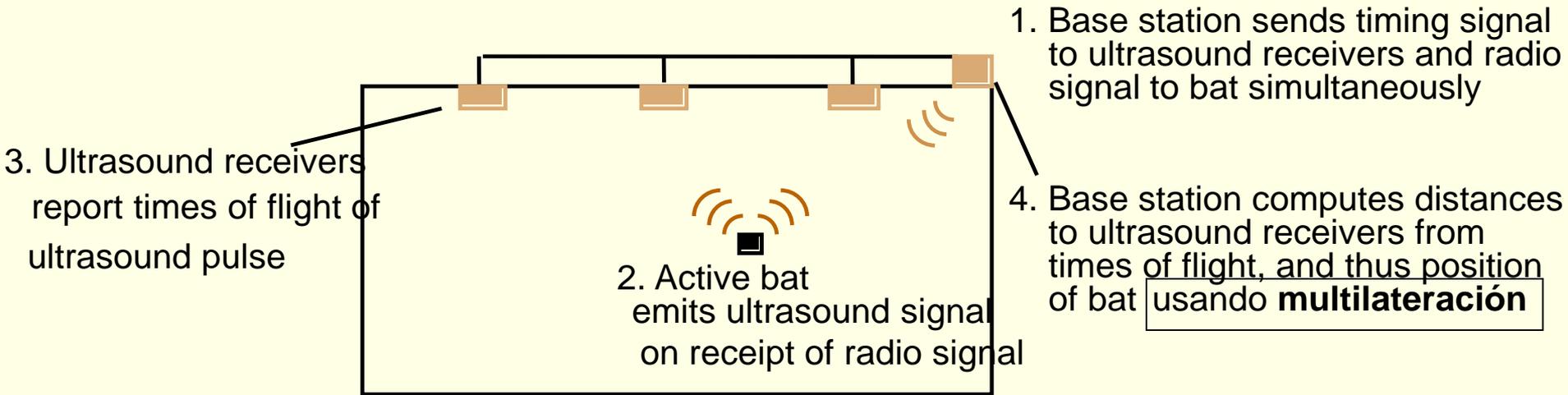
- ☐ Cámara (convencional, infrarrojos)
  - Reconocimiento de objetos reales (o proyectados)
  - Reconocimiento de etiquetas con imágenes
  - Seguimiento de personas (Amazon Go)
- ☐ Luz visible (VLP):
  - permite ToA, RSSI, TDoA, AoA
- ☐ Audio:
  - Normalmente ultrasonidos; OM pasivo o activo
- ☐ Infrarrojos (IR):
  - Proximidad (*Active Badge*)
  - Microsoft Kinect proyecta en estancia patrón IR de puntos
  - Sensor IR detecta profundidad
- ☐ *Inertial Navigation Systems (dead reckoning)*
  - Posición, velocidad y orientación
  - Acelerómetros, giróscopos, magnetómetros
  - Podómetros, efecto Doppler

# Localización basada en UWB



<https://locatify.com/blog/in-practice-precise-indoor-location-detection-with-uwband-ultra-wideband/>

# Active Bat

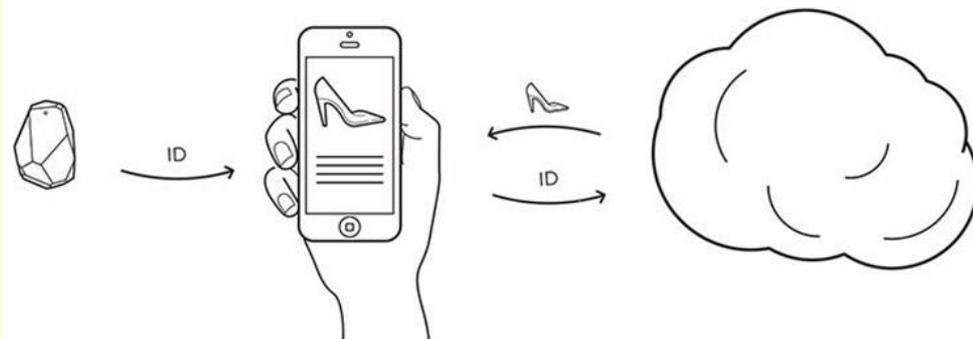


*Distributed Systems: Concepts and Design*  
Coulouris, Dollimore y Kindberg

*UC Fundamentals*  
John Krumm

# *iBeacons*

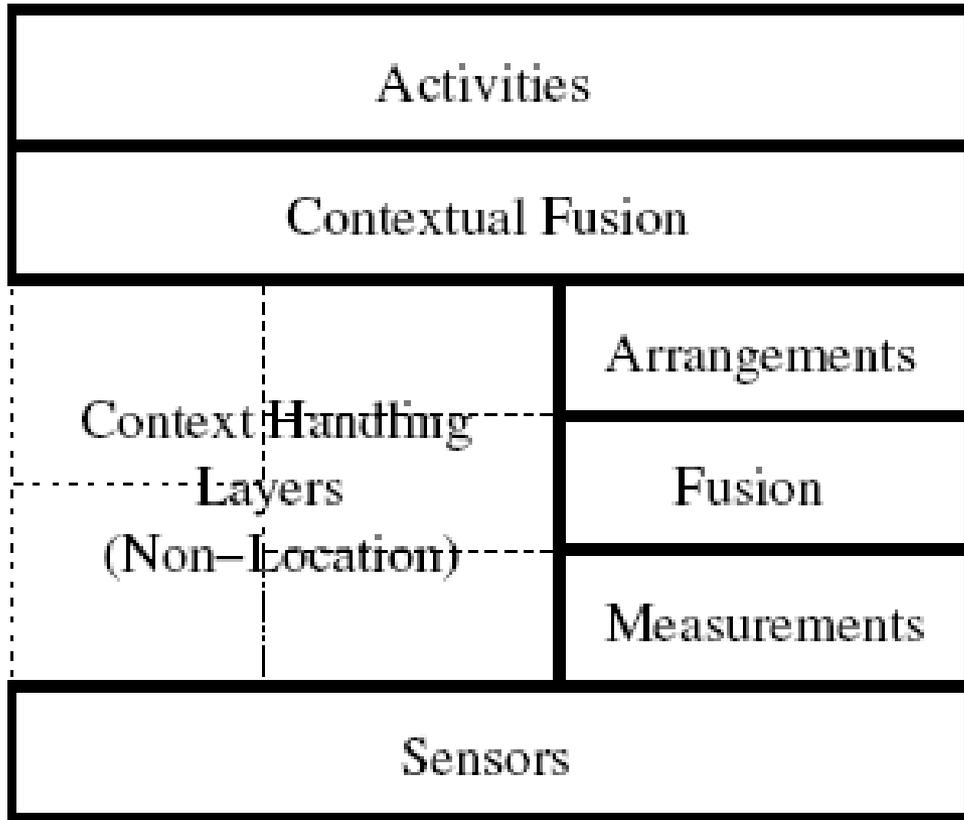
- *beacon broadcast* periódicamente ID usando *BLE*
  - Detección de proximidad: cercanía a posición conocida
  - Por estar en rango señal emitida desde una posición (*iBeacon*)
  - Receptor puede estimar distancia a *beacon* por atenuación RSSI
  - Normalmente, receptor contacta servidor para obtener servicio
    - *mobile marketing* (p.e. anuncios dirigidos)



<http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>

# La pila de localización

Se pueden usar múltiples sensores/técnicas: pila de localización



**Obtiene info. semántica**

**Combina con otra info.**

**Relaciones entre objetos**

**Combina medidas**

**Convertidas a medidas**

**Lecturas de sensores**

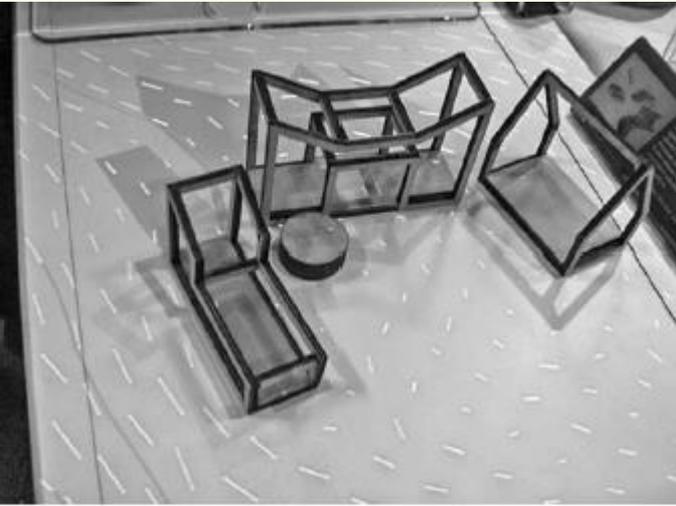
*Location systems for ubiquitous computing*

J. Hightower y G. Borriello

# UI para sistemas ubicuos (UUI)

- ¿Cómo construir IU de sistema que debe quedar en la “periferia”?
- Debe minimizar el nivel de distracción del usuario
  - *Distraction matrix* (Anhalt et al., 2001)
    - Clasifica actividades por grado de atención requerida
      - ▶ *Snap, Pause, Tangent, Extended*
- Múltiples propuestas de IU alternativos:
  - *Tangible User Interface*
  - *Surface User Interface*
  - *Ambient User Interface*
  - *Natural User Interface*
  - .....
- Ejemplo: *The Marble Answering Machine*
  - <https://vimeo.com/183465991>

# Nuevos tipos de IU; extraído de *UC Fundamentals*



**IU tangible**



**IU de superficie**



**IU ambiental**

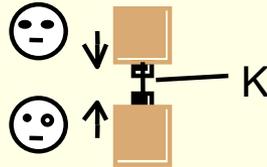
# Distintos tipos de realidad

- ☐ Virtual (VR):
  - Simula mundo real
- ☐ Aumentada (AR):
  - Añade información virtual a mundo real
- ☐ Mixta:
  - *Interpolación* entre RV y RA
  - Hay una gradación continua entre RV y RA
    - Se puede considerar un eje X (*XReality*)
- ☐ Disminuida (DR):
  - Elimina información del mundo real
- ☐ Mediada (generalización de DR)
  - Modifica la información del mundo real presentada

# Seguridad en sistema ubicuos

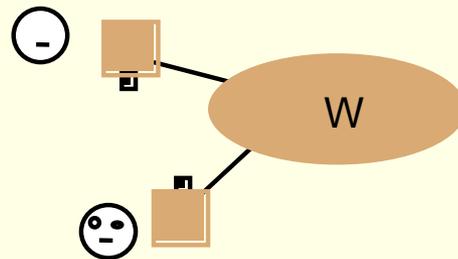
- Potencialmente mucho más inseguros
- Con limitación de recursos
  - Puede descartar uso de cifrado asimétrico
- Sistemas volátiles espontáneos: ¿cómo compartir claves?
  - Comunicación fuera banda por **canal físicamente restringido**
- Misma estrategia que para el problema del límite del EI
  - Mediante contacto o aproximación física entre componentes
    - Uso de comunicación de muy corto alcance o direccional
  - Infrarrojos
  - Audio
  - Láser
  - Código de barras...

# Comunicación segura usando contacto físico



1. Fresh secret key  $K$  exchanged by physical contact

..



2. Devices communicate using secure channel constructed over  $W$  using  $K$

*Distributed Systems: Concepts and Design*

Coulouris, Dollimore y Kindberg

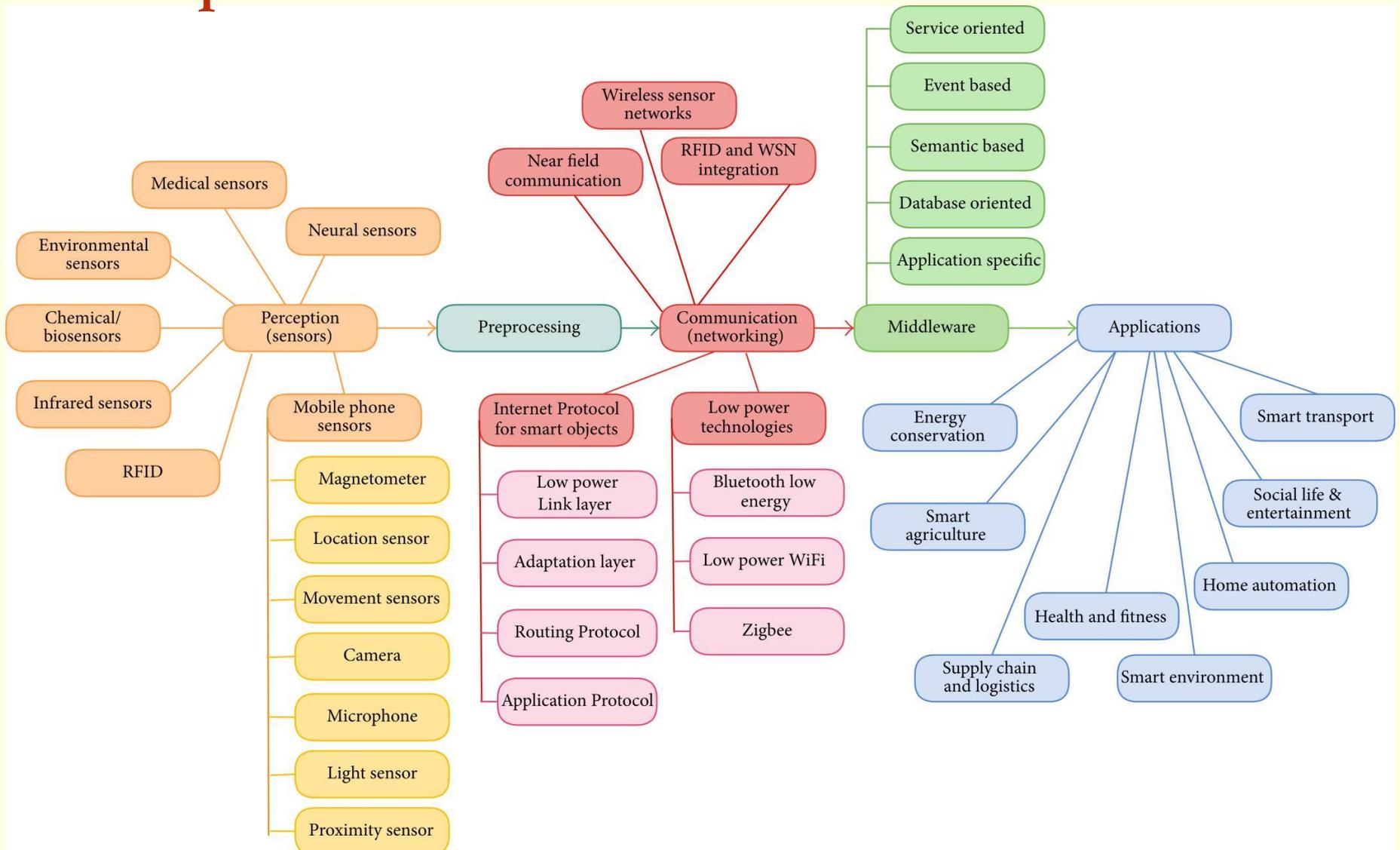
# Aspectos de seguridad en IoT

- Más dispositivos → menos seguridad
  - N° de dispositivos e interconexiones gigantescos
  - Puede afectar a vidas humanas
- Potencialmente, dispositivo inseguro afecta a toda la Internet
- Competencia de mercado
  - Aumenta probabilidad de dispositivos mal diseñados
    - Quizás los costes de un ataque debería pagarlos el diseñador
- Sin elección: Puedo optar por no usar Internet en mi equipo
  - Pero no impedir que me instalen contadores de luz IoT
- Uso masivo: error de diseño en objeto con millones de copias
- Larga vida y dificultad/imposibilidad de actualización
- Invisibles
- Sin IU: ¿Cómo pueden alertar de un problema?

# Aspectos de privacidad en IoT

- ❑ Fuente inmensa de datos de los usuarios
- ❑ No siendo consciente a veces el usuario de su presencia
- ❑ Uso individual de un tipo de información puede no ser relevante
  - Pero sí puede serlo su uso agregado
    - Sensores de taza de café, cepillo de dientes, nevera y fitbit
  - Datos que incluyen además información de ubicación y temporal
- ❑ ¿Cómo implementar “Acepto las condiciones” en dispo. sin IU?
  - Aunque lo tuviera, ¿voy a hacerlo por cada dispositivo?
- ❑ IoT nos iba a traer un mundo más seguro pero no parece que sea así
- ❑ Carácter transnacional:
  - Datos generados en un país y consumidos en otro
- ❑ Reto establecer nueva legislación sobre privacidad en este ámbito

# Recapitulación IoT



**Internet of Things: Architectures, Protocols, and Applications. Pallavi Sethi and Smruti R. Sarangi. Journal of Electrical and Computer Engineering. 2017. <https://www.hindawi.com/journals/jece/2017/9324035/>**