

ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

---

CAMPUS DI CESENA  
SCUOLA DI SCIENZE

Corso di Laurea in Ingegneria e Scienze Informatiche

STUDIO E SPERIMENTAZIONE IN AMBITO WEB  
OF AUGMENTED THINGS

Relazione finale in  
Programmazione di Sistemi Embedded

*Relatore:*

Prof. ALESSANDRO RICCI

*Co-relatore:*

Ing. ANGELO CROATTI

*Tesi di Laurea di:*

MARTINA MAGNANI

---

ANNO ACCADEMICO 2016–2017  
SESSIONE II



# **PAROLE CHIAVE**

**Internet of things**

**Web of things**

**Pervasive computing**

**Augmented reality**

**Web of augmented things**



A me e a mio babbo



# Indice

<b>Introduzione</b>	<b>ix</b>
<b>1 Internet of Things</b>	<b>1</b>
1.1 Tecnologie hardware	5
1.2 Tecnologie e protocolli di comunicazione	9
1.2.1 Wifi	10
1.2.2 RF Links	12
1.2.3 Rete cellulare	12
1.2.4 Bluetooth (BT)	13
1.2.5 Bluetooth Low Energy (BLE)	13
1.2.6 ZigBee	14
1.2.7 Tag RFID (Radio Frequency Identification)	17
1.2.8 NFC	18
1.2.9 iBeacon	20
1.3 Middleware per Internet of Things	21
1.4 Piattaforme per Internet of Things	22
<b>2 Web of Things</b>	<b>25</b>
2.1 Architettura	25
2.1.1 Livello di accessibilità	26
2.1.2 Livello di ricercabilità	28
2.1.3 Livello di condivisione	29
2.1.4 Livello di composizione	29
2.2 Piattaforme per Web of Things	30
2.2.1 EVERYTHING	30
2.2.2 WeIO	31

<b>3</b>	<b>Augmented Reality</b>	<b>33</b>
3.1	Mixed Reality, Virtual Reality e Augmented Reality . . . . .	33
3.2	Tecnologie di visualizzazione - display . . . . .	36
3.2.1	Monitor-based AR display . . . . .	36
3.2.2	See-through AR displays . . . . .	37
3.3	Tracciamento del movimento e relative tecniche di computer vision	42
3.3.1	Optical tracking . . . . .	42
3.3.2	Vision-based tracking . . . . .	44
3.3.3	SLAM . . . . .	49
3.4	Piattaforme per Augmented Reality . . . . .	51
<b>4</b>	<b>Web of Augmented Things</b>	<b>53</b>
4.1	Augmented Worlds - concetti chiave . . . . .	54
4.1.1	Accoppiamento spaziale . . . . .	56
4.1.2	Ologrammi . . . . .	56
4.1.3	Scoperta e osservazione . . . . .	57
4.1.4	Embedding fisico . . . . .	57
4.1.5	Utente . . . . .	57
4.2	WoT + AW: Web of Augmented Things . . . . .	58
4.3	Infrastruttura per WoAT . . . . .	62
<b>5</b>	<b>Sperimentazione dell'idea di WoAT</b>	<b>65</b>
5.1	Definizione del problema . . . . .	65
5.2	Tecnologie . . . . .	66
5.3	Progettazione del software . . . . .	66
5.3.1	Server Rest . . . . .	66
5.3.2	Applicazione client . . . . .	68
5.4	Ipotesi di un framework per WoAT . . . . .	70
5.4.1	Creazione del mondo aumentato . . . . .	72
5.4.2	Creazione dell'applicazione di visualizzazione . . . . .	72
5.4.3	Infrastruttura applicata al caso di studio trattato . . . . .	73
5.5	Valutazione del risultato . . . . .	74
	<b>Conclusioni</b>	<b>77</b>
	<b>Ringraziamenti</b>	<b>79</b>
	<b>Bibliografia</b>	<b>81</b>

# Introduzione

Dall'inizio degli anni '60 è iniziata quella che si potrebbe chiamare seconda "rivoluzione industriale", associata allo sviluppo e alla diffusione delle tecnologie digitali.

Negli ultimi quarant'anni, infatti, si è passati dai mainframe ai personal computer, per arrivare poi nel 2005 all'introduzione del concetto di smartphone che ha portato ad avere un "vero" computer in ogni tasca o borsa. Questo provocò un cambiamento epocale; il computer ha abbandonato la scrivania e ha iniziato ad essere indossato. Oggi, in oltre, le applicazioni più utilizzate sono quelle che offrono servizi che prevedono l'accesso alla rete Internet.

E' proprio la rete Internet che ha permesso un ulteriore passo in avanti, infatti questa viene sempre più utilizzata come una piattaforma in grado di coordinare e far comunicare tra loro oggetti utilizzati nella vita quotidiana. Il progresso delle tecnologie mobile e il supporto di Internet hanno consentito lo sviluppo del concetto di *pervasive computing*, cioè "la computazione ovunque". Con il *pervasive computing*, la percezione fisica del computer scompare, inserendo computazione negli oggetti comuni. Questo approccio nasconde alla persona l'utilizzo diretto del computer e crea una realtà computazionalmente aumentata che permette un'estensione, un *augmentation*, dell'ambiente, delle persone e del mondo.

*Pervasive computing* e rete Internet hanno dato vita a due paradigmi che vengono definiti **internet delle cose** (Internet of Things – IoT) e **realtà aumentata** (Augmented Reality – AR). A loro volta, queste due realtà sono i pilastri per la progettazione degli *augmented worlds* [14].

Internet of Things significa avere qualsiasi oggetto fisico d'uso quotidiano con all'interno capacità computazionale, collegato alla rete internet grazie alla quale può interagire con il resto del mondo. Questo serve per abilitare un insieme di "cose intelligenti", *smart things*, collegate in rete.

L'integrazione degli oggetti intelligenti nel Web ha portato alla definizione di un nuovo concetto che ha assunto il nome di Web of Things (WOT), come descritto

nel “Towards the WOT Manifesto”[1] pubblicato il 10 Aprile 2009.

Con realtà aumentata, invece, si intende quell’insieme di tecnologie che permettono di creare un’estensione della percezione della realtà fisica con l’immersione di oggetti virtuali interagenti fra di loro e con l’ambiente. I livelli aggiuntivi di informazione virtuale devono, perciò, essere accessibili in tempo reale e in modo contestuale, utilizzando opportuni dispositivi di elaborazione e di visualizzazione (come computer portatili, tablets, smartphones e dispositivi wearable). In particolare, lo sviluppo di dispositivi wearable ha permesso di concepire una nuova generazione di sistemi che possono fornire un supporto, insieme alla realtà aumentata, al lavoro quotidiano delle persone: è per mezzo di queste tecnologie che si può iniziare a parlare di una vera e propria *augmentation* delle persone.

In questo lavoro si vuole esaminare l’idea che vede l’applicazione dei principi e dei protocolli del Web of Things anche alla realtà aumentata e che permette, quindi, di avvicinarsi al concetto di Web of Augmented Things (WoAT), come descritto nel “Towards the Web of Augmented Things”[2], pubblicato il \* dal Prof. Alessandro Ricci insieme all’Ing. Angelo Croatti.

In seguito si realizza un software per dispositivi Android che permette sperimentare l’idea di avere oggetti aumentati sul Web.

Il documento sarà composto di quattro capitoli. Nel primo si definiranno gli elementi principali di Internet of Things mentre nel secondo si definirà il concetto di Web of Things. Nel terzo capitolo verrà esaminato il mondo dell’Augmented Reality, definendo alcuni degli elementi base dell’argomento. Nel quarto capitolo, invece, si esporrà la teoria di Web of Augmented Things, mostrando anche un’idea di modello architetturale. Nel capitolo quinto si discuterà della parte di progetto realizzata, esaminando le scelte architettoniche riguardanti la struttura dell’applicazione.

# Capitolo 1

## Internet of Things

Lo sviluppo delle tecnologie digitali sta determinando una progressione esponenziale dei processi di cambiamento in tutti i campi. Tanto che ha sempre più senso e significato parlare di una rivoluzione in atto.

L'impatto della nuova ondata tecnologica è, infatti, vasto e profondo. Dire che ci cambia la vita e il modo di lavorare è giusto, ma rischia di apparire riduttivo; i metodi di comunicazione moderni e l'accesso sempre maggiore a dispositivi tecnologici portatili sono diventati, a partire dai primi anni 2000, parte integrante della vita di chiunque. Questi fattori insieme allo sviluppo di Internet, ora accessibile da qualsiasi parte del pianeta (o quasi) attraverso una miriade di dispositivi diversi, offrono la garanzia di poter disporre di una capacità elaborativa distribuita in grado di soddisfare praticamente qualsiasi richiesta.

E' partendo da questa considerazione che si può introdurre il concetto di pervasive computing, "existing everywhere", cioè l'incorporazione di microcontrollori o microprocessori (nei casi in cui la capacità elaborativa richiesta sia maggiore) all'interno di oggetti di uso quotidiano (everyday objects) portando la computazione "ovunque". Un paradigma computazionale di questo genere nasce proprio grazie allo sviluppo di Internet ed è proprio la rete globale la migliore interfaccia per la comunicazione con il mondo e fra gli oggetti del mondo.

Questi pilastri stanno alla base del paradigma che prende il nome di Internet of Things, l'internet delle cose. Questo termine è stato coniato nel 1999 da Kevin Ashton cofondatore e direttore esecutivo di Auto-ID Center (consorzio di ricerca che risiede al MIT – Massachusetts Institute of Technology).

Essendo un termine recente non ne esiste una definizione precisa e condivisa in letteratura, vorrei quindi proporre alcune:

*”Rete di oggetti dotati di tecnologie di identificazione, collegati fra di loro, in grado di comunicare sia reciprocamente sia verso punti nodali del sistema, ma soprattutto in grado di costituire un enorme network di cose dove ognuna di esse è rintracciabile per nome e in riferimento alla posizione.”* - dall’enciclopedia Treccani.

*“Internet delle cose (Internet of things) è un neologismo riferito all’estensione di Internet al mondo degli oggetti e dei luoghi concreti.”* - da Wikipedia.

*“Oggetti non solo elettronici di uso quotidiano che possono essere riconosciuti, localizzati e controllati attraverso una rete” (NIC 2008)* - dal National Intelligence Council americano (partendo dal concetto di oggetto intelligente).

Passando a definizioni che più si avvicinano al mondo della tecnologia, cito quella fornita dall’IIT [3] (Istituto di Informatica e Telematica) del CNR (Consiglio Nazionale delle Ricerche) che dell’Internet of Things dice:

*“E’ il superamento dei classici limiti della rete che, fuoriuscendo dal mondo virtuale si collega al mondo reale. Tag e sensori infatti, associati a un oggetto, possono identificarlo univocamente e raccogliere informazioni in tempo reale su parametri del suo ambiente.”*

Se ne potrebbero elencare a decine, ma preferisco soffermarmi su quelli che sono i tratti comuni a tutte:

- Presenza di oggetti dotati di interfacce di comunicazione (schede di rete);
- Portabilità elevata di questi oggetti intelligenti e dimensioni ridotte come caratteristica fondamentale;
- Possibilità di localizzazione e identificazione univoca;
- Possibilità di interazione con l’uomo bidirezionale;
- Preponderanza di inserimento capillare di oggetti, sensori e attuatori all’interno dell’ambiente spinta al punto tale che i primi utilizzatori della rete in termini numerici siano macchine e oggetti inanimati;

- Possibilità di proporsi come nuova rivoluzione sociale con risvolti sulle abitudini di vita non ancora ben definiti.

Internet of Things è così recente in primo luogo perché il supporto tecnologico necessario non è banale: se pensiamo che Internet nasce fra gli anni '60 e '70 e le dimensioni delle unità di calcolo dell'epoca erano ingombranti, non di certo nell'ordine dei pochi cm odierni, tutto quello che oggi ci accingiamo a descrivere come una tecnologia consolidata (o per lo meno teoricamente delineata) poteva sembrare un'utopia. In questo arco di tempo sono stati studiati e perfezionati protocolli di comunicazione, diventati poi degli standard, e definite delle tecnologie che hanno permesso di concentrare l'hardware necessario all'elaborazione e alla trasmissione delle informazioni, in dispositivi che possiamo tranquillamente maneggiare, come per esempio gli stessi smartphone, oppure inserire nel contesto ambientale senza che si percepiscano neppure. La cosa più interessante è che questi protocolli, studiati inizialmente per altri scopi, si prestino in modo ottimale ad essere utilizzati, con qualche piccolo perfezionamento, anche dai moderni end device che compongono la parte periferica dell'Internet of Things. Questo è il caso del protocollo TCP/IP e di tutta la famiglia di protocolli dello standard IEEE 802, supportati per l'instradamento e il trasporto delle informazioni all'interno di qualsiasi rete, compresa Internet. Vengono usati XML o JSON per la rappresentazione delle informazioni in formati ben interpretabili da qualsiasi dispositivo ed infine tecnologie come Bluetooth o ZigBee per la comunicazione fra device e device; si può capire quindi che tutto il necessario per l'implementazione delle unità di Internet of Things esiste già e va semplicemente assemblato.

La vera e propria sfida tecnologica risiede in altro. Esistono una serie di aziende diverse che hanno la possibilità di imporsi sulla scena mondiale con l'avvento di questa nuova era dell'informatica e non esistendo veri e propri standard per IoT, ognuno di questi producer ne potrebbe applicare di proprietari. Questo sarebbe certamente una sconfitta, considerando l'obiettivo iniziale prefisso; cioè l'integrazione totale fra mondo fisico e mondo virtuale.

Un'altra considerazione a riguardo si potrebbe fare tenendo conto degli enti privati o pubblici che potrebbero mettere a disposizione finanziamenti per lo sviluppo, questo credo sia un punto centrale per cercare di prevedere la direzione di crescita del settore che potrebbe dirigersi più verso il business e la produzione industriale, oppure verso il controllo, il monitoraggio e la salvaguardia dell'ambiente e della cittadinanza. Sarebbe in ogni caso ottimale definire, per consentire una crescita più veloce e uniforme di questa tecnologia, in che modo essa si debba "costruire" e come debba lavorare in sintonia con ciò che già esiste; con le

altre macchine, con il web e quali debbano essere i limiti di uno sviluppo che rischia di portare ad un controllo troppo invasivo. In particolare dovrà sicuramente essere regolato e standardizzato tutto ciò che riguarda la gestione della sicurezza e della privacy, introducendo opportuni modelli per la gestione dell'identità e del controllo degli accessi.

Esistono inoltre tutt'ora problematiche relative alla tecnologia non trascurabili che possono rallentare lo sviluppo dello scenario applicativo di IoT, prima fra tutte e non trascurabile è l'*indirizzabilità*, ad oggi infatti il protocollo utilizzato è l'IPv4 che, sfruttando 32 bit per l'indirizzamento, non dispone più di indirizzi validi. Il passaggio al protocollo IPv6 è una delle prime sfide tecnologiche che si dovranno vincere per avvicinarsi effettivamente al sogno di un mondo in cui ogni device sia smart in tutto e per tutto.

La *scalabilità* è un'altra sfida per cui il progresso tecnologico dovrà proporre soluzioni per soddisfare il numero di richieste per ogni servizio presente; preché in un mondo che sarà dominato da device intelligenti la richiesta diverrà sempre più elevata. La strada che si sta percorrendo oggi guarda verso l'inserimento di middleware che si frapperanno nella comunicazione user-device, in modo che l'overhead provocato da un numero di connessioni molto elevato non gravi sul dispositivo in sé, in quanto dotato di risorse limitate, bensì su un componente che si inserirebbe nell'architettura con funzionalità (considerando per ora la scalabilità di comunicazione come caso analizzato) simili a quelle di un server web. Anche permettere l'inserimento dinamico di dispositivi all'interno di un'architettura già operativa può non essere banale se si vuole evitare una fase di configurazione iniziale con cui si stabiliscano le modalità e i tempi di comunicazione. Riferendosi nuovamente al middleware, in questo caso come ponte delle comunicazioni tra ogni singolo componente dell'infrastruttura, l'espansione o la riduzione del numero di dispositivi in maniera dinamica sfruttandone le funzionalità può essere molto meno difficoltosa.

Un problema importante riguarda la *tracciabilità* e la scoperta dei servizi presenti nella rete globale: per esempio un utilizzatore di smartphone vuole avere accesso a un sensore di temperatura situato in una precisa località, tralasciando la parte di comunicazione che avviene fra i due, come fa questo utente a trovare ciò che possa soddisfare la sua richiesta? Come fa a trovare esattamente quella risorsa? E' per queste domande e necessità che nasce il concetto di Web of Things ovvero l'integrazione degli oggetti di Internet of Things all'interno, non solo della più grande e importante rete al mondo, ma anche all'interno del WWW – Word Wide Web.

## 1.1 Tecnologie hardware

Le tecnologie utilizzate per la realizzazione degli oggetti intelligenti di Internet of Things derivano da quei dispositivi che in elettronica e informatica si identificano con il termine di sistemi embedded (generalmente tradotto in italiano con “sistema integrato”, letteralmente *immerso* o *incorporato*). Con sistema embedded si definiscono genericamente tutti quei sistemi elettronici di elaborazione digitale a microprocessore progettati appositamente per una determinata applicazione; special purpose, ovvero non riprogrammabili dall’utente per altri scopi, spesso con una piattaforma hardware ad hoc, integrati nel sistema che controllano ed in grado di gestirne tutte o parte delle funzionalità richieste.

Propongo una definizione formale, chiara e semplice del termine “sistema embedded” [4]:

*“una combinazione di componenti sia hardware che software, programmabile o meno, che è progettato per svolgere una funzione specifica o funzioni specifiche all’interno di un sistema più grande.”*

Caratteristiche di un sistema embedded:

- Risulta difficile caratterizzare la velocità o i costi di un sistema embedded generico, anche se, soprattutto per sistemi che devono processare una grande quantità di dati, il progetto stesso assorbe la maggior parte dei costi. Per la maggior parte dei sistemi embedded le prestazioni richieste possono essere soddisfatte con una combinazione di hardware dedicato e una quantità limitata di software ottimizzato.
- La maggior parte dei sistemi embedded è progettata per eseguire ripetutamente un’azione a costo contenuto. La caratteristica di implementare al loro interno un software che ciclicamente viene eseguito li rende riprogrammabili in maniera non troppo complessa. Non richiedono un sistema operativo in quanto la gestione di tutte le risorse è fatta dall’unico software in esecuzione.
- I sistemi embedded hanno tipicamente dimensioni ridotte e sono tipicamente provvisti di batterie oppure tecnologie che gli permettono di immagazzinare energia dall’ambiente. Per questi motivi un fattore importante nella progettazione dei dispositivi è quello dell’efficienza nello sfruttamento di energia. In generale l’efficienza non va mai trascurata per ognuna delle risorse utilizzate, perché quasi sempre limitate.

- I sistemi embedded possono essere fisicamente inaccessibili (come per le batterie di perforazione dei pozzi di petrolio, oppure i componenti lanciati nello spazio), pertanto i sistemi che li contengono devono essere capaci di resettarsi autonomamente in caso di perdita o corruzione dei dati. Questa funzionalità è molto spesso ottenuta con l'inserimento di un componente elettronico chiamato watchdog che riavvia il processore se il programma presente sullo stesso, non azzerava con una certa frequenza il timer interno del componente.
- I materiali con i quali vengono costruiti sono non nocivi e non inquinanti, in quanto i sistemi embedded si trovano spesso immersi nell'ambiente.
- La maggior parte dei sistemi embedded richiede caratteristiche di real-time (o hard real-time), per dispositivi che devono soddisfare compiti dal rigore temporale molto forte.

Degli esempi di sistemi embedded sono: i personal computer dedicati all'automazione industriale e il controllo di processo. I sistemi elettronici per avionica, gli sportelli Bancomat e apparecchi POS, l'elettronica aeronautica, come sistemi di guida inerziale, hardware/software di controllo per il volo e altri sistemi integrati nei velivoli e nei missili, telefoni cellulari e, in generale, dispositivi mobili, centralini telefonici, apparecchiature per reti informatiche come router, timeserver e firewall, switch. Stampanti e Fotocopiatrici. Sistemi di stoccaggio di dati come hard disk, floppy disk o compact disc. Sistemi di automazione casalinghi come termostati, condizionatori e altri sistemi di monitoraggio della sicurezza. Distributori automatici di bevande e cibo, di carburante, di biglietti, ecc.; stazioni automatizzate di riscossione dei pedaggi autostradali o dei parcheggi, ecc. ; in generale distributori o di vari prodotti o servizi. Elettrodomestici come forni a microonde, lavatrici, apparecchi televisivi, lettori o scrittori di DVD, impianti Hi-fi o home video ma anche un banale spazzolino elettrico o un rasoio elettrico di alta fascia. Apparecchiature biomedicali, etc...

Sia per i dispositivi di IoT sia per i sistemi embedded, il componente principale è sicuramente il microprocessore (o microcontrollore). A differenza del mercato dei tradizionali calcolatori, che utilizzano processori con architetture CISC (Complex Instruction Set Computer), i sistemi embedded prediligono processori con architetture RISC (Reduced Instruction Set Computer) più adatte per dispositivi con potenza di calcolo limitata e che richiedono costi ridotti e

basso consumo energetico. Esempi di microcontrollori per sistemi embedded sono: Atmel AVR, Texas Instruments MSP430, Microchip PIC16C84, ARM a 32 bit, Hitachi H8, PowerPC, Espressif ESP8266; tutti nati da produttori diversi a sottolineare il fatto che esiste una frammentazione nella progettazione e nella produzione. Comunque indipendentemente dall'architettura, sono tutti a 8, 16 e raramente a 32 bit. Possiedono dei pin per il collegamento diretto dei dispositivi di input (sensori) ed output (attuatori) e hanno dei pin dedicati alla gestione degli interrupt. Quest'ultimi sono molto importanti durante la fase di progettazione del software, soprattutto nei sistemi non dotati di sistema operativo, perché permettono di segnalare eventi esterni non trascurabili e da processare con urgenza. Le linee di Input/Output sono direttamente programmabili via software, e vengono gestite quando si vogliono utilizzare sensori ed attuatori. I primi sono dei trasduttori, permettono cioè di misurare i fenomeni fisici esterni (come ad esempio la temperatura, la pressione, la luminosità, ...) e convertono la grandezza misurata in segnali analogici o digitali direttamente interpretabili dall'unità di calcolo. I secondi sono degli attuatori, quindi dispositivi che producono effetti misurabili nell'ambiente. Sono questi i meccanismi che permettono al sistema embedded e anche alle unità dell'Internet of Things di interagire con l'ambiente circostante, provando eventi e intercettandone altri.

In alcuni sistemi embedded un aspetto preponderante riguarda la tipologia e relative caratteristiche delle memorie. Ad esempio in sistemi come fotocamere, lettori multimediali, e anche gli stessi smartphone, richiedono l'utilizzo di memorie veloci, silenziose e con un costo non troppo elevato. Date queste necessità, si sta assistendo ad un passaggio sempre più evidente alle memorie di tipo flash.

Le memorie flash sono memorie a stato solido, non volatili e riescono ad avere costi contenuti. Sono efficienti dai punti di vista del consumo e dell'affidabilità. Nei sistemi embedded vengono usate DRAM (Dynamic Random Access Memory) garantiscono una densità maggiore e un minore costo a dispetto delle più veloci SRAM (Static Random Access Memory).

L'evoluzione, che porta a maggiore ingegnerizzazione dei componenti, si sta spostando, specie per sistemi embedded ad alto volume di produzione, verso i "System-on-a-chip" (o SoC).

I SoC racchiudono, in un singolo circuito integrato di tipo ASIC, il microcontroller/CPU e/o il DSP, la memoria, gli oscillatori e il clock, il regolatore di tensione ed eventuali interfacce AD/DA (e verso l'esterno USB, ethernet, etc..). I SoC

sono dotati di una buona flessibilità, evitano al programmatore qualsiasi tipo di progettazione hardware e si rendono subito disponibili all'implementazione del software; fra questo tipo di sistemi il più famoso è sicuramente Arduino.

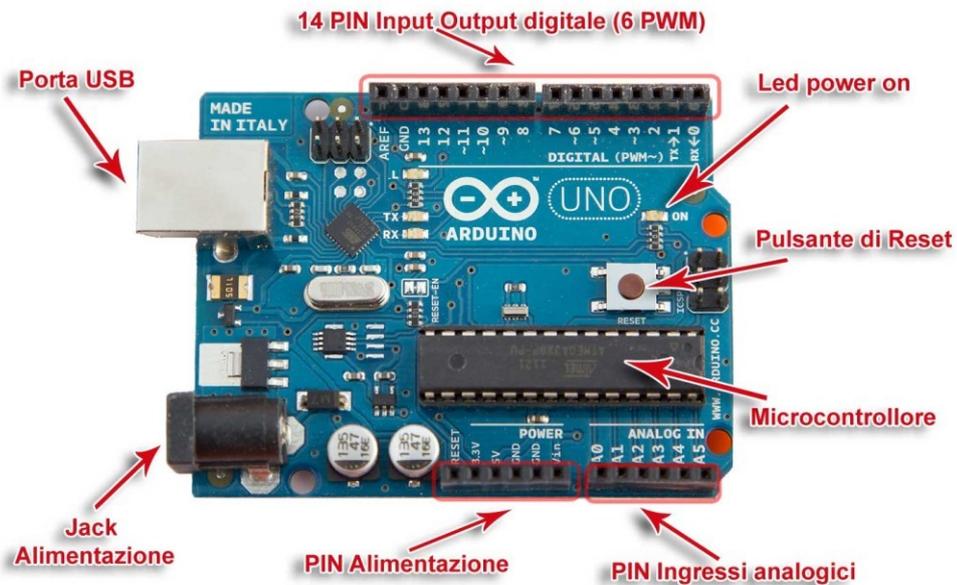


Figura 1.1: Struttura e componenti del SoC Arduino UNO

Le “cose” o “oggetti” dell’Internet of Things sono oggetti fisici che dispongono di un identificativo univoco, di un sistema embedded e l’abilità di trasferire dati nella rete. Quindi l’insieme di questi oggetti intelligenti è in pratica un’interconnessione di dispositivi computazionali embedded univocamente identificati che interagiscono e comunicano mediante Internet. A questo punto mi piacerebbe citare Nicholas Negroponte (cofondatore del MIT) che ha detto: *“When we talk about an Internet of Things, it is not just putting RFID tags on some dumb things so we smart people know where that dumb thing is. It’s about embedding intelligence so things become smarter and do more than they were proposed to do”* (Nicholas Negroponte, MIT)

## 1.2 Tecnologie e protocolli di comunicazione

Per iniziare a parlare di comunicazioni tra dispositivi è necessario introdurre un nuovo concetto che è quello delle Machine To Machine (M2M), facendo riferimento a quell'insieme di tecnologie che supportano la comunicazione wired e wireless tra dispositivi/macchine al fine di permettere uno scambio di informazioni locali, ad esempio la temperatura e lo stato del dispositivo.

Sono tecnologie utilizzate per lo più in funzioni di monitoraggio e controllo. In particolare sono state utilizzate in applicazioni di telemetria e automazione industriale.

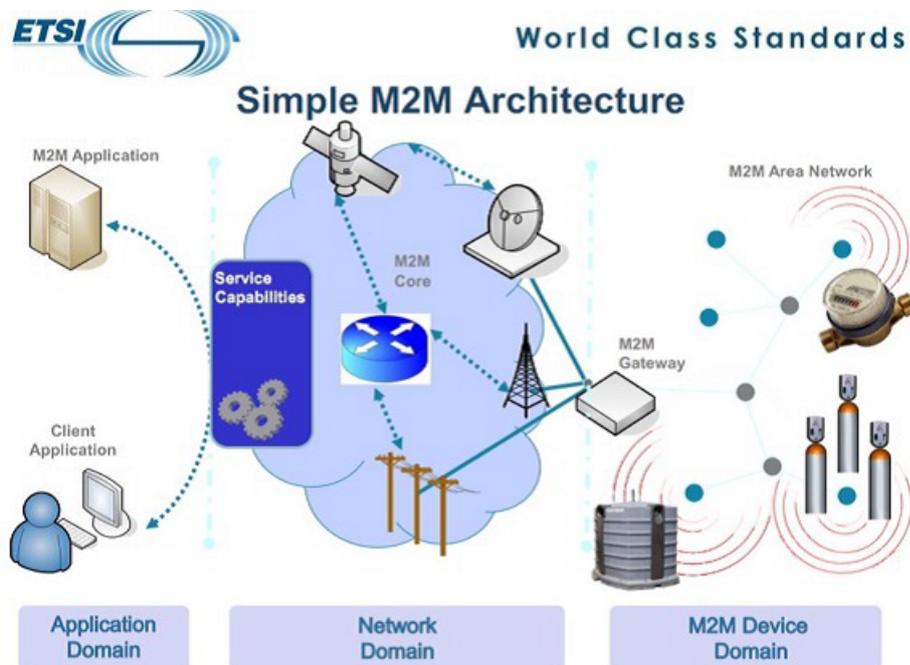


Figura 1.2: Esempio di comunicazione fra dispositivi utilizzando un'architettura M2M.

Oggi giorno M2M è considerata parte integrante dell'Internet of Things, infatti si fa riferimento a reti di dispositivi in grado di acquisire informazioni dall'ambiente (mediante sensori), computare e quindi agire producendo effetti (attuatori), connessi mediante varie tecnologie di rete che permettono loro di interagire in modo cooperativo.

La comunicazione fra dispositivi è un aspetto cruciale di M2M e Internet of Things, ovvero la capacità di scambiare informazioni via rete fra dispositivi e sottosistemi (embedded e non).

La comunicazione può essere sia wired, sia wireless, in entrambi i casi i dispositivi vengono dotati di opportuni moduli di comunicazione.

Nel caso wired tipicamente sono shield/schede con porta Ethernet e protocollo 802.11. Nel caso wireless, possono essere shield oppure dongle USB, contenente il modulo con ricevitore/trasmittitore e relativa antenna.

La comunicazione fra microcontrollore e moduli di comunicazione avviene usualmente mediante la porta seriale.

Le tecnologie e i protocolli wireless si possono suddividere in base alle tipologie di applicazioni in cui sono più appropriati e maggiormente utilizzati:

Applicazioni generali:

- Wifi
- RF Links
- Rete cellulare

Nei sistemi Embedded:

- Bluetooth/BLE
- ZigBee

In sistemi che servono apposite tecnologie radio per l'identificazione:

- RFID
- NFC
- IBeacon

### **1.2.1 Wifi**

Il Wi-Fi è una tecnologia per la creazione di reti senza fili basata sullo standard IEEE 802.11 (quindi sulla famiglia di protocolli che usano TCP/IP a livello di trasporto), la sua portata varia in base al dispositivo e può passare da pochi metri a centinaia rendendolo perfetto per creare infrastrutture di rete in collegamento ad Internet.

Può operare in due modalità: modalità infrastrutturata e modalità ad hoc.

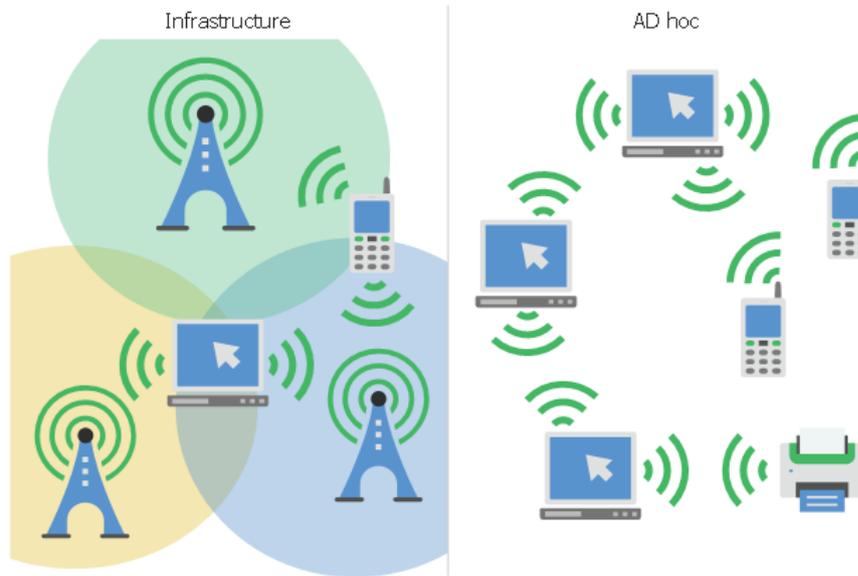


Figura 1.3: Modalità infrastrutturata del Wi-Fi e modalità AD hoc del Wi-Fi.

- Modalità ad hoc: viene detta “ad hoc” perché non si basa su un’infrastruttura preesistente, come i router in reti cablate o punti di accesso in reti wireless (infrastrutture). Invece, ogni nodo partecipa al routing inoltrando i dati verso altri nodi, quindi la determinazione dei nodi a cui inoltrare i dati viene effettuata dinamicamente sulla base della connettività di rete e dell’algoritmo di routing in uso [5].

Le reti ad hoc wireless sono “self-configuring”, quindi reti dinamiche in cui i nodi sono liberi di muoversi.

Le reti wireless non dispongono delle complessità dell’installazione e dell’amministrazione delle infrastrutture, consentendo ai dispositivi di creare e unire le reti “on the fly”, “in volo”: ovunque e in qualsiasi momento [6]. Una rete wifi ad hoc non necessita di un punto di accesso e ogni membro della rete può comunicare con gli altri scambiandosi informazioni, questo può creare molti problemi in quanto la larghezza di banda è divisa per il numero di host e tutta la connessione rallenterebbe in presenza di un host

più lento degli altri, in quanto la velocità a cui viaggiano i dati ha un limite superiore.

- **Modalità infrastrutturata:** un access point raccoglie tutti i dati trasmessi da ogni singolo host e solo questo access point può mandare informazioni ad un altro host. La larghezza di banda rimane così preservata, e per reti con un numero di membri elevato è possibile utilizzare più di un solo punto di accesso in modo che un dispositivo possa comunicare con l'access point più vicino diminuendo il tasso di interferenze presente nelle connessioni. Aumentando l'affidabilità della rete, si rispetta uno di quei principi sui quali qualsiasi buona architettura IoT si deve basare.

In generale questo tipo di tecnologia, come le altre wireless, offre il vantaggio di non essere legata a nessun tipo di cablaggio dei cavi permettendo di inserire sensori in luoghi molte volte non raggiungibili altrimenti, d'altra parte la richiesta di energia è molto elevata e il sovraffollamento della banda destinata a questo standard può rendere le comunicazioni difficili in ambiti in cui la presenza di reti Wi-Fi domestiche è cospicua. I costi sono contenuti sia per i moduli che permettono la comunicazione sia per il fatto che le connessioni sullo spettro di banda dedicato allo standard IEEE 802.11 sono libere da licenza.

Il wifi è utilizzato anche in ambito embedded sebbene non sia stato progettato allo scopo, il consumo è rilevante ma può essere gestito.

### **1.2.2 RF Links**

Sono interfacce semplici di comunicazione via radio. Sono caratterizzate da Data rate fino a 1 Mbps, e coprono fino a 1 Km a seconda dell'antenna, con una frequenza di 2,4 GHz. Non supportano protocolli di alto livello come TCP/IP. Non sono adatte all'internet of things.

### **1.2.3 Rete cellulare**

La comunicazione avviene mediante rete dati cellulare broad-band. L'evoluzione degli standard va dal GPRS, al 3G, al WiMax fino al 4G. Hanno una copertura nell'ordine dei Km ma consumi elevati, per questo non sono ideali per IoT.

### 1.2.4 Bluetooth (BT)

Nelle telecomunicazioni il Bluetooth è uno standard tecnico-industriale di trasmissione dati per reti personali senza fili (WPAN: Wireless Personal Area Network). Fornisce un metodo standard, economico e sicuro per scambiare informazioni tra dispositivi diversi attraverso una frequenza radio sicura a corto raggio. Un dispositivo può ricercare ogni altro dispositivo che offra lo stesso tipo di comunicazione all'interno del raggio di copertura delle onde radio (qualche decina di metri). Questi dispositivi possono essere ad esempio palmari, telefoni cellulari, personal computer, portatili, stampanti, fotocamere digitali, smartwatch, console per videogiochi purché provvisti delle specifiche hardware e software richieste dallo standard stesso. Il BT si è diffuso da tempo anche nel settore industriale (strumenti di misura, lettori ottici, ecc) per il dialogo con i relativi datalogger.

Ogni dispositivo Bluetooth è in grado di gestire simultaneamente la comunicazione con altri 7 dispositivi sebbene, essendo un collegamento di tipo master-slave, solo un dispositivo per volta possa comunicare con il master. Questa rete minimale viene chiamata piconet. Il master è il dispositivo che all'interno di una piconet si occupa di tutto ciò che concerne la sincronizzazione del clock degli altri dispositivi (slave) e la sequenza dei salti di frequenza. Gli slave sono unità della piconet sincronizzate al clock del master e al canale di frequenza.

Le specifiche Bluetooth consentono di collegare diverse piconet in modo da espandere la rete. Tale configurazione di rete viene chiamata scatternet.

Gli slave possono appartenere a più piconet contemporaneamente, mentre il master di una piconet può al massimo essere lo slave di un'altra. Il limite di tutto ciò sta nel fatto che all'aumentare del numero di piconet aumentano anche il numero di collisioni dei pacchetti e di conseguenza degradano le prestazioni del collegamento.

### 1.2.5 Bluetooth Low Energy (BLE)

Il Bluetooth essendo uno standard di successo, nato nel 1999 grazie a una cooperazione fra le più importanti aziende del settore elettronico tra cui Sony Ericsson, IBM, Intel, Toshiba e Nokia, ha avuto diverse evoluzioni nel tempo che hanno garantito via via sempre prestazioni maggiori con un consumo di energia sempre minore. Vista poi l'effettiva adattabilità di questa specifica ai dispositivi di dimensioni e potenza ridotta si è pensato di sviluppare BLE (Bluetooth Low

Energy, a volte per questioni commerciali chiamato anche Bluetooth Smart) che opera alle stesse frequenze e con le stesse modalità del normale standard BT ma si rende perfetto per l'integrazione in dispositivi quali beacon, orologi da polso, tastiere, giocattoli o sensori per lo sport. Il Bluetooth è determinante per l'effettiva realizzazione di Internet of Things[InfoServiziLocalizzazioneApple] e a tal proposito riporto le parole di un'intervista rilasciata da Suke Jawanda, capo marketing del Bluetooth Special Interest Group:

*"We can credibly say that Bluetooth Smart technology is what has made the Internet of Things a reality. It's no coincidence that the flood of wearables we're seeing come to market today started flowing after Bluetooth Smart was integrated into iOS and iPhone 4s in late 2011. Since then, it's become the de facto standard for fitness devices, smart watches and medical devices from companies as diverse as Nike, Polar, Adidas, Pebble, Samsung, Qualcomm, Sony, iHealth, and Nonin. And with all major mobile operating 14 1. Internet of Things systems now offering native Bluetooth Smart APIs, this will only accelerate as manufacturers can create products and associated apps that will run on the smartphones, tablets and PCs that people already own. Going beyond wearables, there are a number of wireless solutions that will help to enable the IoT, but because Bluetooth Smart is standards based, inexpensive and already shipping in billions of devices every year, it's going to be a major part of making the IoT a practical reality. Manufacturers know Bluetooth Smart can help them bring new products to huge markets quickly. We're already seeing this as Bluetooth Smart is being widely adopted in beacons for retailing and payment systems, and growing fast in smart home applications as well."*

Lo standard BLE si basa sulla versione 4.1 del protocollo di cui si possono individuare tre qualità ben precise: usabilità, flessibilità e capacità di adattarsi al mondo di IoT.

### **1.2.6 ZigBee**

In telecomunicazioni, nel mondo delle tecnologie wireless ZigBee rappresenta uno dei principali standard di comunicazione, curato dalla ZigBee Alliance.

Lo standard specifica un insieme di protocolli di comunicazione che basano il proprio funzionamento su antenne radio digitali a bassa potenza, si stima infatti che una rete di sensori che comunica grazie a questa tecnologia possa garan-

tire, con la stessa batteria, le proprie funzionalità per circa due anni; grazie a questa caratteristica, al costo contenuto e all'assenza di vincoli nell'utilizzo, ha preso piede soprattutto in ambito industriale e nel settore della domotica. E' una tecnologia di tipo wireless, recente e molto utilizzata nei sistemi embedded per creare delle Wireless Personal Area Network (WPAN) in base allo standard IEEE 802.15.4. Questa tecnologia è nata con lo scopo di essere più semplice e più economica di altre Wireless PAN come, ad esempio il Bluetooth (quest'ultima per ricorrere ai ripari ha successivamente rilasciato una versione meno dispendiosa in termini di consumi, il Bluetooth Low Energy).

Ci sono tre differenti tipi di dispositivo ZigBee:

- **ZigBee Coordinator (ZC):** è il dispositivo più "intelligente" tra quelli disponibili, costituisce la radice di una rete ZigBee e può operare da ponte tra più reti. Ci può essere un solo Coordinator in ogni rete, esso è in grado di memorizzare informazioni riguardo alla sua rete e può agire come deposito per le chiavi di sicurezza.
- **ZigBee Router (ZR):** questi dispositivi agiscono come router intermedi passando i dati da e verso altri dispositivi.
- **ZigBee End Device (ZED):** includono solo le funzionalità minime per dialogare con il suo nodo parente (Coordinator o Router), non possono trasmettere dati provenienti da altri dispositivi, sono i nodi che richiedono il minor quantitativo di memoria e quindi risultano spesso più economici rispetto ai ZR o ai ZC.

Topologie di rete utilizzate con ZigBee:

- Topologie star e peer-to-peer: ogni rete deve avere almeno un ZED che funge da coordinatore (ZC).
- Topologie cluster tree: struttura in cui un ZED può avere un solo ZC.
- Topologie mesh: struttura in cui vi sono molti nodi che comunicano fra loro. Da ogni nodo è possibile comunicare con qualsiasi altro nodo. Infatti è molto affidabile, nel caso in cui un nodo venga meno per problemi hardware o altro, nodi vicini semplicemente cercano altri percorsi per trasmettere il segnale (si rivolgono ad altri nodi).

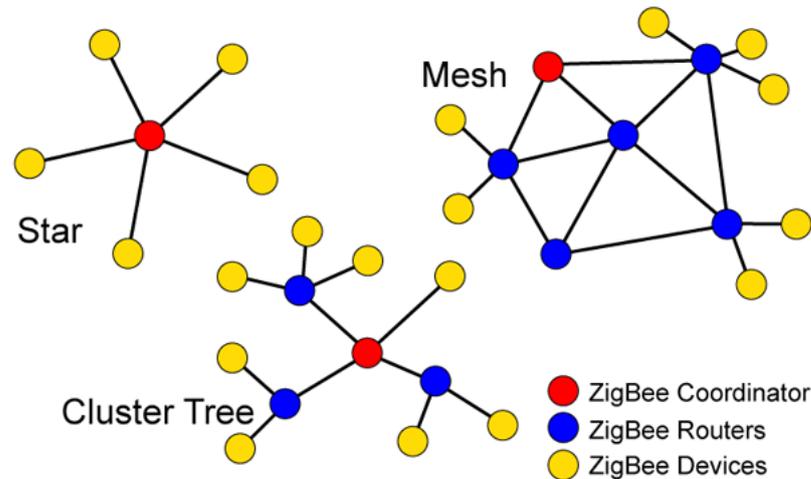


Figura 1.4: Topologie di rete ZigBee.

ZigBee permette di organizzare la rete ed ottimizzarne il controllo. Qualsiasi dispositivo si voglia unire ad essa deve informare il coordinatore, che fornirà a quest'ultimo un indirizzo di rete valido che verrà impiegato nei pacchetti di rete al posto del MAC Address.

Ci sono nodi router sempre in attesa di ricevere chiamate ed ogni pacchetto che circola per la rete può essere consegnato solo attraversandone uno; quando un router riceve il pacchetto ne rileva la destinazione e verifica se il nodo sia attivo oppure in stand-by. Nella prima ipotesi procede con l'inoltro del pacchetto mentre altrimenti avvia un ciclo di attesa. L'utilizzo di questo protocollo per connettere in rete dispositivi intelligenti è molto indicata perché si tratta dell'unica tecnologia wireless specificatamente concepita e progettata per fornire soluzioni di rete a basso costo, per sensori a bassa potenza e per il controllo di network che includono una grande varietà di dispositivi.

La possibilità di collegarne fino a 65.000 dispositivi in un'unica rete, rende ZigBee per ora imbattibile e la soluzione migliore per IoT.

Le qualità di ZigBee lo rendono ideale per collegare gli oggetti d'uso quotidiano, per controllare i dispositivi elettronici di consumo, per gestire l'impiego di energia, per aumentare il comfort dell'utente ed infine per la sicurezza.

### 1.2.7 Tag RFID (Radio Frequency Identification)

In telecomunicazioni ed elettronica con l'acronimo RFID (Radio-Frequency IDentification, in italiano "identificazione a radiofrequenza") si intende una tecnologia per l'identificazione e/o memorizzazione automatica di informazioni inerenti oggetti, animali o persone, grazie a particolari etichette elettroniche, chiamate tag (o anche transponder o chiavi elettroniche e di prossimità) e alla capacità di queste di rispondere all'interrogazione a distanza da parte di appositi apparati fissi o portatili, chiamati reader (o anche interrogatori).

L'RFID è una tecnologia di comunicazione a bassa potenza importantissima per il mondo IoT, rende infatti possibile l'integrazione di un metodo di comunicazione wireless all'interno di dispositivi a bassissima potenza e di dimensione molto ridotte come per esempio una calamita, determina un consumo di energia praticamente nullo ed è quindi ottimale per essere usato in ogni caso in cui non si abbia accesso a fonti di energia a lunga durata.

Un tag RFID contiene tipicamente un ID univoco che contraddistingue l'oggetto su cui è montata l'etichetta, ed è possibile leggere l'id grazie ad opportuni reader. Dal punto di vista fisico un tag è costituito semplicemente da un chip, un'antenna e un involucro protettivo e può essere distinto in tre tipologie: tag passivo, tag attivo o tag BAP (battery assisted passive).

- Il tag passivo è di piccole dimensioni e non è alimentato, per questo il costo è molto ridotto. E' caratterizzato da un circuito che viene alimentato indirettamente, a distanza, dalle onde radio inviate da un RFID reader che ne vuole leggere il contenuto (tipicamente l'ID). Può anche essere scritto da un RFID reader, usando lo stesso principio.
- Il tag attivo ha una batteria interna, le dimensioni sono maggiori rispetto a quelli passivi, e trasmettono in broadcast il proprio contenuto informativo.
- I battery assisted passive (BAP) sono tag ibridi, presentano caratteristiche simili ai tag attivi ma invia le proprie informazioni solo se sollecitato da un RFID reader. Supporta distanze maggiori rispetto ai tag passivi.

Il reader è un ricetrasmittitore che contiene una CPU, parti digitali di I/O per consentire il collegamento seriale con un computer e componenti di elettronica analogica per la trasmissione e ricezione dei dati. E' dotato di una fonte di alimentazione autonoma.

Le onde radio utilizzate hanno un range molto ampio che va dai 30 kHz ai 3GHz

per le microonde, in base alla frequenza di trasmissione si individuano diversi standard definiti da organizzazioni quali l'Auto-ID Center e l'ISO. Per i tag più piccoli si utilizzano frequenze nell'ordine dei kHz (Low Frequency) che permettono delle comunicazioni a distanza di centimetri, da 3MHz a 30MHz si parla di High Frequency che hanno un range di decine di centimetri; questo range di frequenze è quello utilizzato nello standard Near Field Communication (NFC) ISO 14443.

Per comunicazioni a raggio più ampio vengono invece utilizzate delle frequenze più elevate, come UHF (Ultra-High-Frequency) e Microonde che arrivano fino a 3GHz. In queste situazioni il consumo di potenza comincia ad essere elevato. Questi reader vengono utilizzati, ad esempio, all'interno dei dispositivi di lettura dei pedaggi autostradali in velocità.

Gli attori nella comunicazione RFID sono: il reader, il tag e un middleware che funge da interfaccia tra applicazione e reader fisico. Il middleware si occupa di interpretare le richieste provenienti dal software applicativo e di comandare il reader. Viceversa, i dati letti dal reader vengono solitamente riportati verso l'applicazione.

Il middleware può utilizzare uno schema di collegamento diretto al reader, fornendo i comandi appropriati direttamente sulla porta di comunicazione oppure si può appoggiare a librerie del produttore dell'hardware che istanziano un reader virtuale con il quale comunicare tramite chiamate API. L'utilità del middleware risiede nella capacità di riuscire a comandare hardware di tipologia differente rendendo quindi il software applicativo svincolato dai componenti effettivamente utilizzati.

### **1.2.8 NFC**

Near Field Communication (NFC) (in italiano letteralmente "Comunicazione in prossimità") è una tecnologia che fornisce connettività wireless (RF) bidirezionale a corto raggio (fino a un massimo di 10 cm). È stata sviluppata congiuntamente da Philips, LG, Sony, Samsung e Nokia.

Questa tecnologia è evoluta da una combinazione della tecnologia RFID e altre tecnologie di connettività. Contrariamente ai più semplici dispositivi RFID, gli NFC permettono una comunicazione bidirezionale. Quando due apparecchi NFC (il target e il initiator) vengono accostati entro un raggio 4 cm, viene creata una rete peer-to-peer tra i due, ed entrambi possono inviare e ricevere informazioni.

Il target può essere dato da semplici tag, come nel caso di RFID.

Questa tecnologia non è recente, ma negli ultimi anni è diventata molto utilizzata nel settore degli smartphone. Infatti, integrando chip e antenna NFC nello smartphone è possibile realizzare sistemi di pagamento (avvicinando lo smartphone a degli appositi terminali nei negozi), sistemi di condivisione di dati tra dispositivi, accedere a contenuti digitali leggendo gli smart tags (piccoli chip di sola lettura che si possono trovare in poster e documenti, come badge o passaporti), sistemi di autenticazione (accedere a reti senza dover inserire password e tanto altro).

La semplicità con cui questa tecnologia permetta di “trasformare” uno smartphone in una carta di credito senza fili, permette di intuire la sua potenzialità.

Il funzionamento è il seguente: il dispositivo che inizia la comunicazione usa un’antenna per emettere un segnale radio a 13.56 MHz, generando quindi un campo elettromagnetico. Se un dispositivo NFC ‘è presente nel campo, viene attivato e quindi creato un canale di comunicazione. La comunicazione dovrà però avvenire solo con un destinatario, perciò in caso di presenza di più dispositivi si potrà comunicare solo con uno alla volta. La comunicazione si dice attiva se ognuno degli “interlocutori” (l’iniziatore della comunicazione e il destinatario) genera un proprio campo, si dice passiva se a generare il campo è solo colui che ha iniziato la comunicazione. Negli NFC la comunicazione è half-duplex e il bit rate può essere 106, 212 o 424 kbps, quindi non molto veloce, perciò vanno scambiate piccole quantità di dati alla volta.

Un dispositivo NFC si può utilizzare in tre modalità diverse:

- **Lettore/scrittore:** in questa modalità, il dispositivo NFC può leggere e modificare dati contenuti in tag NFC passivi (che si possono trovare ad esempio nei poster), permettendo all’utente di recuperare delle informazioni. Per quanto riguarda i consumi energetici, l’energia per generare il campo NFC deve essere fornita da un dispositivo attivo.
- **Emulazione di carta:** un dispositivo NFC può comportarsi come una smart card. Una smart card è una carta tascabile con dei circuiti integrati (solitamente memoria volatile e microprocessore). In questa modalità un lettore RFID esterno non può effettuare distinzioni tra una carta e un dispositivo NFC. Una smart card può essere emulata sia a livello applicativo sia usando un componente chiamato Secure Element (dispositivo simile alle vere smart card). Questa modalità è utile per i pagamenti elettronici.
- **Peer-to-peer:** questa modalità permette a due dispositivi NFC di stabilire una connessione bidirezionale per scambiare dati. Si distingue qui l’inizia-

tor, cioè colui che fa la richiesta (client) e il target (host), cioè colui che riceve la richiesta e risponde. L'host non può iniziare la comunicazione se prima non è stato interpellato dall'initiator. L'energia richiesta per creare il campo NFC è condivisa tra il target e l'initiator.

### 1.2.9 iBeacon

iBeacon è un sistema di posizionamento indoor descritto da Apple come [6] “una nuova classe di trasmettitori, a bassa potenza e a basso costo, che possono notificare la propria presenza a dispositivi vicini con iOS7 e iOS8”. Questa tecnologia è stata introdotta recentemente per fare localizzazione e identificazione wireless in ambienti indoor. La tecnologia consente ad uno smartphone o ad un altro dispositivo di effettuare delle azioni quando sono nelle vicinanze di un iBeacon. Sfrutta la tecnologia BLE (Bluetooth Low Energy), cioè utilizza la percezione di prossimità del Bluetooth Low Energy per trasmettere un identificativo unico universale (UUID), che sarà poi letto da una specifica app o sistema operativo. Una volta letto il segnale, l'app può svolgere varie azioni programmate come accedere ad un social network o inviare una notifica push. Nello specifico un iBeacon è un piccolo trasmettitore a bassissimo consumo che emette continuamente un segnale contenente il suo ID, che può essere rilevato da un qualsiasi dispositivo con BLE.

I messaggi sono composti da:

- **UUID (Universally Unique Identifier):** stringa di 16 bit usata per differenziare tra di loro i gruppi di beacon diversi. Ad esempio se un'azienda ha una rete di beacon, questi avranno lo stesso UUID.
- **Major:** stringa di due bit usata per distinguere sottogruppi all'interno del gruppo più grande.
- **Minor:** stringa di due bit che serve per identificare univocamente il singolo beacon.

Un possibile scenario applicativo può essere il seguente: un utente si trova all'interno di un negozio, e un'applicazione dedicata sul suo smartphone si mette a captare beacon. A seconda dei dati captati, l'utente può essere notificato con informazioni che possono tornargli utili, come offerte e promozioni speciali. Questa tecnologia sta prendendo molto spazio nel mondo dell'IoT.

### 1.3 Middleware per Internet of Things

In informatica con middleware si intende un insieme di programmi informatici che fungono da intermediari tra diverse applicazioni e componenti software.

Una definizione tecnica, reperibile sul sito dell'università italiana di Roma: La Sapienza, è testualmente:

*"... un software di connessione che consiste in un insieme di servizi e/o di ambienti di sviluppo di applicazioni distribuite che permettono a più entità (processi, oggetti, ecc.), residenti su uno o più elaboratori, di interagire attraverso una rete di interconnessione a dispetto di differenze nei protocolli di comunicazione, architetture dei sistemi locali, sistemi operativi, ecc. "*

La sua funzione è quella di dare organizzazione e fornire delle interfacce che nascondano ai livelli superiori l'implementazione e l'organizzazione dei livelli sottostanti. Gli svantaggi dell'assenza di un middleware sono facilmente individuabili: assenza di un meccanismo di protezione degli accessi, assenza di scalabilità, di interoperabilità e assenza di un'interfaccia comune per l'accesso agli smart device. Il middleware in ambito Internet of Things non viene introdotto solo per garantire un'interfaccia di utilizzo generale utilizzabile da risorse differenti, ma deve fornire inoltre:

- **Connettività e Comunicazione:** il middleware deve garantire interoperabilità a livello di comunicazione fra i dispositivi della rete, per questo in molte architetture si preferisce far passare ogni singola comunicazione attraverso il nodo principale, evitando connessioni dirette fra due peer della rete. In questo ai peer è sufficiente conoscere al più un protocollo di rete, dato che tutte le comunicazioni sono gestite dal middleware. E' il middleware a conoscere ogni protocollo usato da ogni dispositivo per la comunicazione, in questo modo è in grado di fare la traduzione dei messaggi per renderli comprensibili in modo del tutto trasparente. Se questo non avvenisse invece le possibilità di comunicazione sarebbero limitate oppure bisognerebbe adeguare ogni singolo componente a "parlare la stessa lingua", questo limiterebbe molto le possibilità di espansione delle reti di sensori.
- **Gestione dei dispositivi:** il middleware deve sempre avere consapevolezza del contesto in cui opera e questo comporta la necessità di gestire ogni minima azione compiuta dai dispositivi. Con il termine contesto intendiamo lo stato di ogni dispositivo, le sue funzioni, la sua sicurezza, il suo SW e

così via. Quindi sicuramente fra le principali funzioni di device management troviamo: scoperta dinamica dei dispositivi; eliminazione dalla rete di dispositivi difettosi; abilitare e disabilitare funzionalità HW; aggiornamento firmware e SW da remoto; localizzazione dispositivi compromessi o smarriti; reset delle informazioni non più sicure

- Raccolta dei dati, analisi e attuazione: i device al di sotto del middleware non dispongono, per motivi di costi, dimensioni ed efficienza, di supporti adeguati per la memorizzazione dei dati, quindi tutte le informazioni rilevate dai dispositivi sono collezionate in apposite strutture dati di alto livello memorizzate nel middleware oppure (negli approcci più moderni) utilizzando il Cloud distribuito.
- Scalabilità: come già anticipato quello della scalabilità per le reti di sensori è un problema molto importante. Quindi il middleware viene interposto come un proxy che permette la gestione di tutte le connessioni entranti, smistandole secondo la capacità di ogni singolo sensore, garantendo affidabilità maggiore all'intero sistema.
- Sicurezza: questo è uno degli aspetti più critici quando si parla di Internet of Things. Oltre ai problemi tradizionali che esistono per ogni rete di calcolatori, si vengono a creare delle ulteriori problematiche dovute alle tecnologie specifiche utilizzate in questo campo. Prendiamo come esempio la tecnologia RFID: essa, tramite il meccanismo dei reader, è potrebbe facilmente subire attacchi di intercettazione, modifica o interruzione. Durante la trasmissione del segnale lettori non autorizzati potrebbero captarlo. Inoltre esiste un altro tipo di attacco, detti "Tag tampering", in cui persone malintenzionate potrebbero cambiare il contenuto stesso di uno specifico tag [19]. Altri inconvenienti sono legati all'autenticazione e al riconoscimento: sarebbe buona prassi non utilizzare solo username e password, ma gestire la procedura mediante protocolli con meccanismi di token, cioè di dispositivi hardware personali dotati di generatori pseudocasuali di numeri che si presentano sotto forma di oggetti comuni come portachiavi.

## 1.4 Piattaforme per Internet of Things

Lo scopo di una piattaforma per Internet of Things è quello di collegare un insieme di sensori a reti di dati e fornire approfondimenti attraverso applicazioni

di backend sui dati generati dai sensori. Sono disponibili molte piattaforme per IoT che forniscono la possibilità di distribuire applicazioni on the go, ovvero che permettono di monitorare in qualsiasi momento lo stato della propria rete di sensori utilizzando dispositivi mobili, quali smartphone, tablet e così via.

La differenza tra una piattaforma e un middleware è che una piattaforma consiste in un insieme di strumenti, prodotti e standard utilizzati in combinazione per fornire una soluzione ad un problema; un middleware è un componente di tale soluzione. Due esempi di piattaforme per IoT sono:

- **Amazon Web Services IoT Platform (AWS IoT)**; AWS IoT è una piattaforma cloud gestita che consente a dispositivi connessi di interagire in modo semplice e sicuro con applicazioni nel cloud e altri dispositivi. Ciò consente di raccogliere dati di telemetria da più dispositivi e memorizzarli o analizzarli. È inoltre possibile creare applicazioni che consentono agli utenti di controllare questi dispositivi dai loro smartphone o tablet. AWS IoT è in grado di supportare miliardi di dispositivi e migliaia di miliardi di messaggi, ed è in grado di elaborare e instradare tali messaggi agli endpoint di AWS e ad altri dispositivi in modo sicuro e affidabile. Con AWS IoT, le applicazioni rimangono collegate e comunicano con tutti i dispositivi, in qualsiasi momento, anche quando non sono collegati.
- **Android Things**; Android Things (nome in codice Brillo) è un sistema operativo embedded basato su Android, è stato annunciato per la prima volta al Google I/O 2015. È stato progettato per essere utilizzato da dispositivi IoT a basso consumo di energia e con bassa quantità di memoria RAM (32-64 MB), che sono spesso costruiti partendo da differenti piattaforme per microcontrollori. Avrà il supporto per il Bluetooth Low Energy e il Wi-Fi ed utilizzerà il protocollo Nest Weave2 per la comunicazione tra dispositivi. Nest Weave è un protocollo di rete introdotto da Nest nel 2015 per fornire comunicazioni sicure, robuste e affidabili tra i dispositivi collegati.



# Capitolo 2

## Web of Things

Web of Things (WoT) è un termine usato per indicare un'estensione di IoT, in cui non si vede più solamente un dispositivo intelligente connesso in rete ma anche immerso, a livello di applicazione, nel World Wide Web. Nel WoT gli oggetti fisici di uso quotidiano vengono incorporati nel Web dando loro una API, questo facilita la creazione di profili virtuali degli oggetti e la loro integrazione e riutilizzo in tipi diversi di applicazioni. Questa idea nasce nel 2007 quando alcuni ricercatori, tra i quali Dominique Guinard e Vlad Trifa, hanno pubblicato il manifesto "Towards the WOT Manifesto"[1], nel quale si propone di utilizzare gli standard del web per creare il livello applicativo di Internet of Things. Non viene creato nessun nuovo standard e nessuna nuova tecnologia, bensì viene riutilizzato e adattato tutto ciò che già esiste per realizzare applicazioni web; tecnologie come HTTP, Web Socket, JSON, REST, etc...

Viene incoraggiato l'utilizzo di Server Web con un'architettura REST, in quanto permettono la creazione di servizi facilmente riutilizzabili. REST è attualmente il nucleo del Web.

*"Rest, REpresentational State Transfer, è un'architettura di rete per sistemi distribuiti basati sull'ipertesto, esso delinea come ogni risorsa all'interno del web sia definita e indirizzata."* (Representational State Transfer, Wikipedia).

### 2.1 Architettura

Dato che ancora non è stato creato un standard, il "web delle cose" si basa su un insieme di best practice che possono essere classificate in base all'architettura

Web of Things[8, 9].

L'architettura propone quattro livelli principali (o fasi) che vengono utilizzati come un framework per classificare i diversi modelli e protocolli coinvolti.

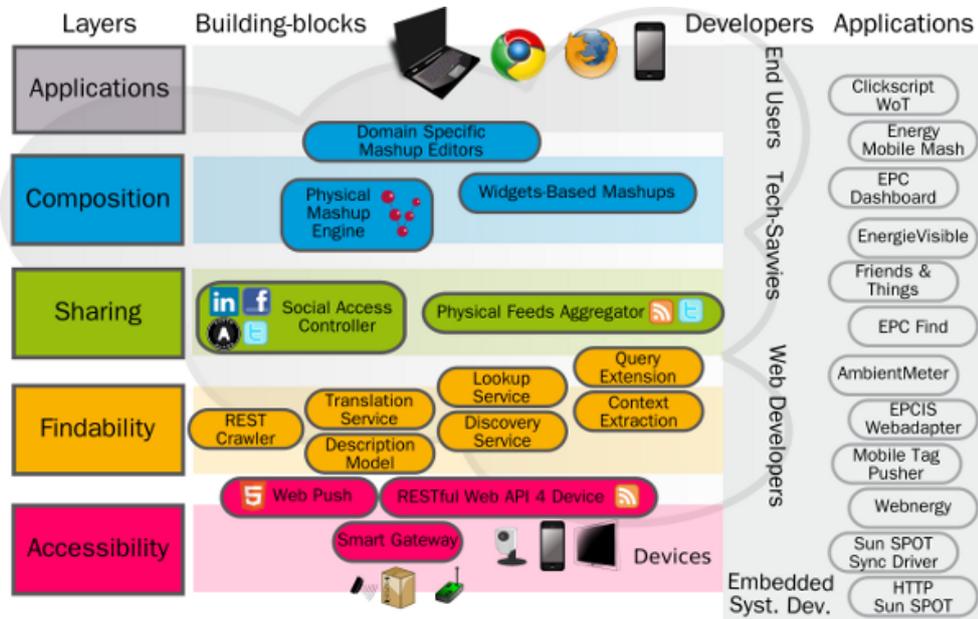


Figura 2.1: Tecnologie ed applicazioni dei livelli che compongono l'architettura di Web of Things.

### 2.1.1 Livello di accessibilità

Questo livello si occupa di far accedere le cose/oggetti ad Internet ed esponendo i servizi di ognuno tramite API Web. E' il livello principale del WoT, in quanto assicura che ogni oggetto fisico disponga di una propria API accessibile via Web, trasformandoli così in oggetti programmabili [8, 9, 10].

Questo livello viene costruito sulla base di due patterns: in primo luogo, tutte le "cose" devono esporre i loro servizi attraverso un'API RESTfull (direttamente o tramite gateway). In secondo luogo, la natura di richiesta-risposta di http è spesso citata come una delle limitazioni, in quanto non corrisponde alla natura delle applicazioni comuni alle reti di sensori wireless.

Per ovviare a questa discordanza, mantenendo l'attenzione sull'integrazione con il Web, diversi autori hanno suggerito l'utilizzo di HTML5 WebSockets in modo

nativo, o tramite l'utilizzo di un broker di traduzione (es: traducendo da MQTT o CoAP a WebSockets). Questo permette di integrare l'API REST degli oggetti ad un meccanismo di sottoscrizione/pubblicazione tipico dell'ecosistema Web.

### Rest

REpresentational State Transfer (REST) è un tipo di architettura software per i sistemi di ipertesto distribuiti come il World Wide Web. L'espressione "representational state transfer" e il suo acronimo "REST" furono introdotti nel 2000 nella tesi di dottorato di Roy Fielding, uno dei principali autori delle specifiche dell'Hypertext Transfer Protocol (HTTP), e vennero rapidamente adottati dalla comunità di sviluppatori su Internet.

Questa tecnologia si differenzia dalle altre per la creazione di servizi web perché non utilizza nessun livello opzionale per il proprio funzionamento: si appoggia ad http per trasmettere i dati sfruttando tutte le opzioni che tale protocollo mette a disposizione; questo non avviene ad esempio nel protocollo SOAP (Simple Object Access Protocol) che pur rientrando nella categoria degli standard adibiti alla chiamata remota di procedure, sfrutta semplicemente il protocollo HTTP per far trasportare dati che rispettino la formattazione tipica XMLWSDL (Web Services Description Language) e sono inclusi nel payload dei pacchetti.

Con la nascita di applicazioni RESTful all'interno del web si è iniziato a parlare di Web 2.0, detto anche web dinamico, per sottolineare il fatto che un approccio di questo tipo ha aumentato l'interazione fra utenti e applicazioni online. A differenza del web statico, il quale offriva semplicemente una consultazione di documenti, permette agli utenti l'accesso a servizi quali blog, social network ed uso interattivo di piattaforme di e-commerce.

L'idea centrale di applicazioni REST risiede nel fatto di catalogare qualsiasi entità sotto forma di risorsa; oltre a questo naturalmente esistono altre prerogative che devono essere rispettate:

- Il web deve essere basato sull'identificazione univoca delle risorse attraverso indirizzi URI (Uniform Resource Identifiers).
- Ogni risorsa deve essere disponibile attraverso un'interfaccia che ne riassume le funzionalità. HTTP fornisce un insieme di comandi (GET, PUT, DELETE, POST, HEAD, etc.) che possono essere utilizzati per dialogare con una risorsa; questo insieme di comandi, se usato con una semantica ben precisa, è sufficiente per concludere qualsiasi tipo di comunicazione.

ne con ogni risorsa. Per qualsiasi tipo di comunicazione si intendono le tipiche operazioni CRUD (Create, Read, Update, Delete).

- La comunicazione avviene con messaggi auto-esplicativi che permettono di eliminare la fase di negoziazione individuale del formato. In quest'ottica sono molto utilizzati i formati JSON e XML che oltre ad essere molto leggeri, offrono una perfetta integrazione con i linguaggi di scripting più comuni quali JavaScript e PHP.
- La comunicazione fra client e server deve essere priva di stato: la richiesta deve contenere al suo interno tutte le informazioni che il server necessita per l'interpretazione e per fornire una risposta adeguata; anche se si fa uso di meccanismi quali cookies per informare il server dell'identità del client, non obbliga i due attori della comunicazione a gestire una particolare sessione. La principale ragione di questa scelta è la scalabilità: mantenere lo stato di una sessione ha un costo in termini di risorse sul server e all'aumentare del numero di client tale costo può diventare insostenibile. Inoltre, con una comunicazione senza stato è possibile creare cluster di server che possono rispondere ai client senza vincoli sulla sessione corrente, ottimizzando le prestazioni globali dell'applicazione.

REST, fornendo queste linee guida, non specifica mai come debba avvenire l'effettiva implementazione e lascia libero il programmatore di utilizzare poi la tecnologia che reputa essere la migliore; ad oggi per sviluppare applicazioni di questo tipo la tecnologia AJAX (Asynchronous JavaScript and XML) è la più usata.

### 2.1.2 Livello di ricercabilità

La famiglia dei protocolli facenti parte di questo livello si occupa di fornire algoritmi per ricercare particolari servizi o particolari oggetti all'interno del vasto mondo del web.

Questo livello ha la peculiarità di offrire tutti quegli strumenti che possano far diventare ogni risorsa non solo accessibile da qualsiasi entità facente parte del web, ma anche ricercabile e catalogabile da altre applicazioni.

L'approccio è quello di riutilizzare gli standard semantici del Web per descrivere le "cose"/oggetti e i loro servizi. In particolare, le persone hanno lavorato sull'integrazione di HTML5 Microdata, RDF/RDFa, JSON-LD e EXI. Ciò consente di cercare le cose attraverso i motori di ricerca e gli altri indici Web, oltre

a consentire ad una macchina di interagire con altre macchine basandosi su un piccolo set di formati e standard ben definiti.

### 2.1.3 Livello di condivisione

Uno dei motivi principali per cui si integrano i dispositivi intelligenti all'interno del web è quello di rendere accessibile al mondo intero una grandissima mole di dati, per permettere di migliorare le abitudini e lo stile di vita. A questo proposito lo sharing layer integra i dati con strumenti e applicazioni web di più alto livello che rendono l'analisi dei Big Data più efficiente, veloce e soprattutto sicura. E' a questo livello che vengono implementate le politiche di autenticazione degli utenti e di protezione dei dati secondo le norme del codice privacy.

Una soluzione per la condivisione delle informazioni partendo da un qualcosa che già esiste sono i social network: lo sviluppo in tema di privacy, avuto da tutte le piattaforme social negli ultimi anni, garantisce meccanismi altamente affidabili e non facilmente violabili per la protezione delle informazioni e la condivisione con le sole persone che rientrano all'interno della "rete" personale di ogni utente. Tutti i social network utilizzano delle strutture a grafo per la memorizzazione delle relazioni di ogni utente, essi grazie alla loro diffusione globale costituiscono uno spaccato molto realistico su quelli che sono i rapporti esistenti tra le persone e l'introduzione di un livello di autenticazione social al di sopra dei dispositivi intelligenti porta a definire il concetto di Social Web of Things[10].

### 2.1.4 Livello di composizione

In questo livello si cerca di avvicinare gli sviluppatori alla creazione di applicazioni Web 2.0 che utilizzino i dati e le funzionalità fornite dalla rete di sensori globale, qui il focus si sposta maggiormente verso la parte di front-end e il livello di presentazione dei dati.

Nel web dinamico assume grossa importanza il concetto di aggregazione intesa come aggregazione di contenuti e funzionalità provenienti da fonti esterne (API, RSS o JavaScript) a livello di presentazione; nel tempo questo meccanismo ha preso sempre più piede e di conseguenza sempre maggiore 'e il numero di tool (Google App Engine o Yahoo Pipes) messi a disposizione per avvicinare anche i meno esperti alla sua realizzazione, 'e possibile idearne uno senza scrivere nemmeno una riga di codice. L'alternativa ai tool è rappresentata sempre dall'utilizzo delle interfacce REST o SOAP fornite da un applicativo web attraverso linguaggi

di programmazione come PHP, ASP.NET o Java.

Per comprendere meglio cosa si intende per mashup prendiamo come esempio i maps mashup, molto utilizzati nel web moderno: essi sono mashup dove i dati di una determinata fonte vengono visualizzati all'interno di mappe; per questo scopo vengono molto usati i servizi di Google Maps. Quest'ultimo permette la visualizzazione di mappe di tutto il mondo ed offre l'accesso libero alla piattaforma Maps, quindi a tutti i suoi servizi e a tutte le web API, affiancati da un'ottima documentazione con lo scopo di incentivare gli utenti a sviluppare sempre nuovi servizi.

## 2.2 Piattaforme per Web of Things

### 2.2.1 EVERYTHING

EVERYTHING è una piattaforma dedicata agli sviluppatori che vogliono utilizzare le funzionalità del Web of Things, fondata nel 2011 da Niall Murphy e Andy Hobsbawm. Essa si occupa di gestire l'identità digitale dei dispositivi intelligenti collegati nel web e fornisce supporto per la creazione di applicazioni real-time che utilizzino tali dispositivi.

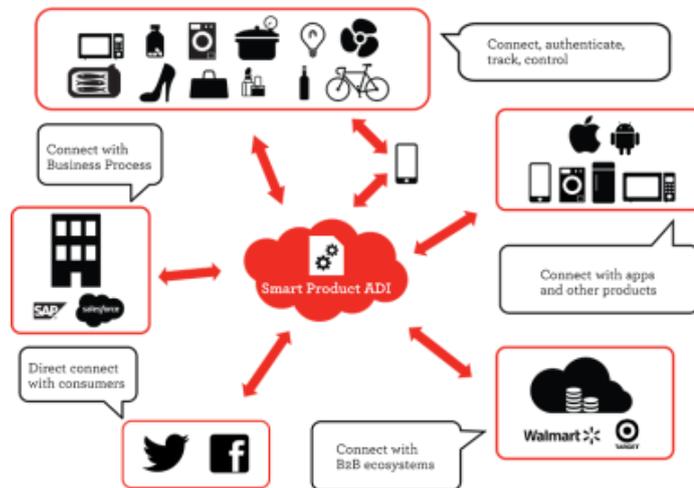


Figura 2.2: Integrazione degli Active Object Identifier in oggetti di uso quotidiano.

All'iscrizione vengono forniti la documentazione e gli strumenti necessari per inserire i propri dispositivi intelligenti all'interno del web. Ad ogni oggetto viene assegnato un ADI (Active Digital Identity), cioè un identificatore univoco all'interno del Cloud dell'organizzazione. Alla fine di questa operazione l'oggetto fisico è diventato effettivamente un Web Object utilizzabile fruttando le API per sviluppatore a disposizione.

L'Active Digital Identity di una things è in grado di fornire descrizioni standard mediante metadati che consentono alle applicazioni di comprenderne la semantica e di utilizzarne i servizi. I dispositivi fisici, essendo in questo modo trattati come Web Object, possono essere facilmente collegati ad altre risorse web ed entità come social network, record di CRM (Customer Relationship Management) [10], Sistemi ERP (Enterprise Resource Planning) [12], etc. L'identità online può essere rafforzata generando QRcode e tag NFC specifici per rappresentare l'URI della risorsa nel web in modo univoco. Gli ADI permettono inoltre di catalogare gli oggetti per funzionalità dinamiche o statiche e di generare in maniera non molto complessa, integrazioni con alcuni tra i più famosi servizi online; come Flickr o Google Maps.

Per capire meglio cosa si intende per identità digitale e Web Object vediamo il seguente esempio: per effettuare la registrazione di un oggetto presso l'account di uno sviluppatore, avviene uno scambio di messaggi HTTP fra client e piattaforma. Quando l'oggetto viene registrato, lo sviluppatore ne diventerà il proprietario con la possibilità di gestire utilizzando appositi tag, tutte le configurazioni che riguardano la *ricercabilità* nel web. Inoltre può decidere quali siano i diritti di utilizzo degli altri utenti. Ogni applicazione prodotta da terzi, dopo essersi registrata e se in possesso della chiave di accesso può utilizzare i servizi dei dispositivi attraverso le API sviluppatore.

Il progetto collegato a questa piattaforma ha riscontrato un notevole successo e non sono poche le aziende leader del settore tecnologico che hanno mostrato interesse: dal 2011 (anno di lancio) ad oggi, gli investimenti sono nell'ordine di milioni di dollari e la base operativa si è estesa dall'Europa all'America contando registrazioni di prodotti in tutti e cinque i continenti.

### 2.2.2 WeIO

WeIO6 è una piattaforma hardware e software open source per la prototipazione e la creazione di oggetti interattivi connessi (con collegamenti senza fili) utilizzando solo linguaggi popolari nell'ambito della programmazione web. Quindi

HTML5, CSS3 e JavaScript per la programmazione lato client e Python 2.7 per la programmazione lato server.

WeIO utilizza come implementazione del meccanismo Zeroconf [13] Apple Bonjour; tale meccanismo consente di individuare automaticamente la presenza di dispositivi WeIO nella rete locale. E' possibile accedere al file system di ogni dispositivo WeIO attraverso SSH [14] e Samba. Per programmare un dispositivo bisogna collegarsi al dispositivo desiderato attraverso un web browser e utilizzare il web editor integrato per scrivere il codice sorgente. Un altro metodo è quello di scrivere il codice sorgente sulla propria macchina e copiarlo successivamente all'interno della memoria del dispositivo.

## Capitolo 3

# Augmented Reality

Lo straordinario sviluppo delle tecnologie sta riducendo il limite tra materia fisica e digitale, tra realtà e virtualità. La nozione di *augmentation* può fare riferimento a diversi concetti riguardanti in primo luogo la realtà aumentata, in secondo luogo l'*environment augmentation* relativo a Internet/Web of Things (come abbiamo visto nei precedenti capitoli) e infine la *human augmentation* supportata dalle tecnologie wearable.

Negli ultimi anni il dibattito sulla realtà aumentata è stato molto acceso e ha fomentato l'interesse del grande pubblico, però ha anche portato confusione sulle definizioni principali dell'argomento.

### 3.1 Mixed Reality, Virtual Reality e Augmented Reality

La mixed reality (MR), a volte riferita come “hybrid reality”, è la fusione di mondi reali e virtuali per produrre nuovi ambienti e visualizzazioni in cui gli oggetti fisici e digitali coesistono e interagiscono in tempo reale. Il primo sistema di mixed reality è stata la piattaforma Virtual Fixtures sviluppata negli Armstrong Laboratories dell'Aeronautica, degli Stati Uniti d'America nei primi anni '90. Nel 1994 Paul Milgram e Fumio Kishino definiscono la mixed reality come “...*anywhere between the extrema of the virtuality continuum*”[17], dove il “virtuality continuum” si estende dall'ambiente completamente reale, all'ambiente completamente virtuale attraverso la coesione tra realtà aumentata (AR) e realtà virtuale (VR). Secondo Milgram il mondo reale e quello virtuale rappresentano

due condizioni estreme.

Come si vede dalla Figura 3.1, la realtà aumentata si trova vicino all'ambiente reale, in quanto sono gli oggetti digitali ad essere immersi in esso, mentre la realtà virtuale si trova vicino all'ambiente virtuale, in quanto sono gli oggetti reali ad essere immersi nel mondo virtuale.

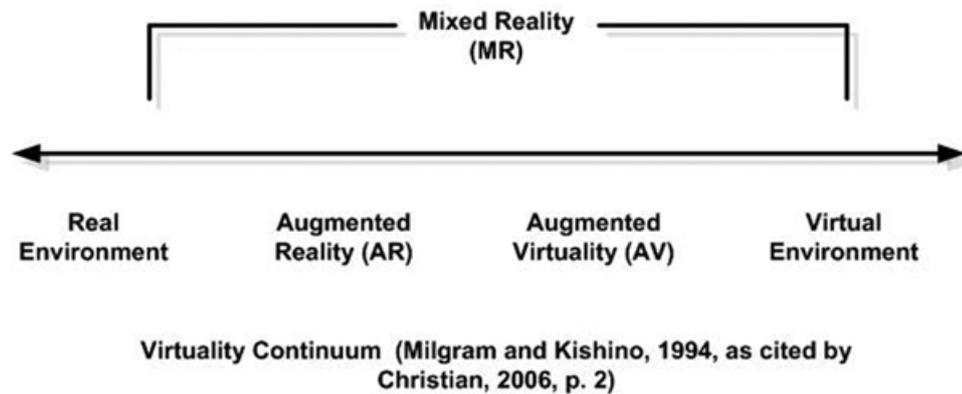


Figura 3.1: Distribuzione delle tecnologie nel *virtuality continuum*.

Realtà aumentata e realtà virtuale hanno un'importante caratteristica in comune: entrambe hanno la capacità di alterare la nostra percezione del mondo. Ciò che le differenzia, è la tipologia di alterazione.

In particolare, la realtà virtuale descrive esperienze in cui l'utente è interamente immerso in un mondo virtuale tridimensionale e interagisce con esso. Si tratta di una simulazione computer-generated che fa sentire l'utente come se stesse vivendo quella nuova realtà in prima persona. Questa simulazione viene percepita da tutti i suoi sensi; per primi la vista e l'udito. La paternità del concetto è poco chiara, VR come termine è generalmente attribuito a Lanier [26] che ha lavorato attivamente a questo dominio alla fine degli anni '80.

La realtà virtuale è definita da almeno tre proprietà cardini [18]:

- **immersione:** l'utente deve avere la sensazione di essere parte della realtà virtuale che viene rappresentata. La percezione di immersione può essere mentale nel caso in cui l'utente immerga se stesso nell'ambiente attraverso il pensiero. Questo è quello che succede tipicamente quando si guarda un'immagine tridimensionale sullo schermo. Oppure, l'immersione può essere anche fisica nel caso in cui, attraverso strumenti sofisticati di rappresentazione dell'immagine come occhiali stereoscopici associati a cuffie

stereofoniche (stereoscopic head mounted displays) e datagloves, l'utente abbia la sensazione di essere parte dell'ambiente di realtà virtuale ricevendo allo stesso tempo stimoli visivi, acustici e tattili;

- **navigazione:** rappresenta la capacità di muoversi all'interno dell'ambiente virtuale esplorando elementi e strutture da diversi punti di vista, senza alcuna restrizione. Questa caratteristica rende possibili anche importanti applicazioni in ambito terapeutico e chirurgico in cui possono essere esaminati gli organi da un punto di vista interno;
- **interazione:** implica la capacità di interagire con la realtà virtuale in tempo reale manipolando e trasformando gli elementi come se fossero reali. Questo rappresenta uno degli aspetti più complessi dell'intero sistema e richiede un'enorme capacità di calcolo da parte dei computer ed algoritmi che devono essere in grado di prevedere diverse possibilità d'interazione da parte dell'utente con il sistema.

La realtà aumentata, invece, sebbene possa essere pensata come una variazione della realtà virtuale, ha una differenza sostanziale: non immerge l'utente in un mondo virtuale, ma sovrappone oggetti virtuali al mondo reale percepito dall'utente. Essenzialmente può essere vista come "l'aggiunta di "cose" virtuali sopra il mondo reale percepito".

La realtà aumentata lascia all'utente la possibilità di interagire in tutto e per tutto con il mondo reale, anche se quest'ultimo viene arricchito con elementi virtuali che possono essere percepiti dai sensi umani come rappresentazioni di oggetti 2D/3D, suoni e odori che si fondono con oggetti fisici o con l'ambiente reale: dunque gli elementi virtuali non rimpiazzano la realtà, bensì cooperano con la stessa fornendo informazioni che l'utente altrimenti non potrebbe direttamente rilevare con i propri sensi.

Ronald Azuma, Capo di ricerca presso Centro Ricerche Nokia, infatti, afferma che "*Augmented Reality (AR) is an environment that includes both virtual and real-world elements*" [19], quindi lo scopo non è quello di sostituire il mondo fisico, come tende a fare la realtà virtuale, ma di estendere la realtà.

La particolarità della realtà aumentata è combinare e registrare il mondo reale con un modello digitale del sistema. L'ambiente reale e virtuale sembrano coesistere e l'utente si può muovere liberamente nella scena, con la possibilità, altresì, di interagire con essa. Tutto ciò deve essere ovviamente elaborato in maniera ottimale, ovvero, in maniera tale che l'utente abbia la percezione di una singola

scena, nella quale le entità reali e virtuali siano pressoché indistinguibili. I campi di utilizzo della realtà aumentata sono molteplici, alcuni esempi riguardano l’ambito militare, medico, manifatturiero e culturale/artistico.

Per consentire esperienze di augmented reality, si possono utilizzare diversi tipi di display. Di seguito si introdurranno alcune delle classi principali e si forniranno esempi concreti.

## 3.2 Tecnologie di visualizzazione - display

### 3.2.1 Monitor-based AR display

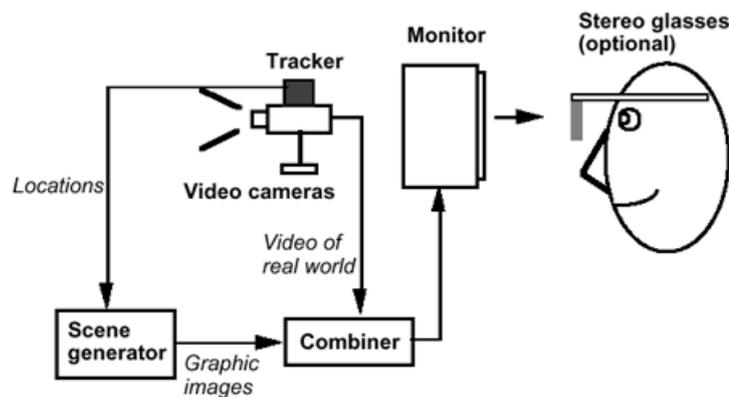


Figura 3.2: Principio di funzionamento dei display AR monitor-based.

I display AR monitor-based (Figura 3.2) sono la tipologia più semplice; si riferiscono a esperienze non immersive, dove l’aumento si verifica su uno schermo “lontano”, come ad esempio un televisore o uno smartphone (anche se in questo caso, viene utilizzato il palmare più consone alle esigenze richieste). Il display è trattato come una finestra sul mondo aumentato (da qui il nome “Window-on-the-World”, definito da Milgram [20]), creato da video immagini in diretta o registrate.

A causa della sua accessibilità (da punto di vista dell’utente) è la più prominente forma di esperienza AR, attuabile in molteplici applicazioni, ad esempio nelle trasmissioni sportive (Figura 3.3). Anche il gioco Pokémon Go, che ha riscos-

so un grande successo, è un esempio di monitor-based AR, svolgendosi sulle schermate degli smartphone (Figura 3.4).



Figura 3.3: Strumento “AR SkyPad” usato da Sky Sports per analizzare le azioni dei giocatori.

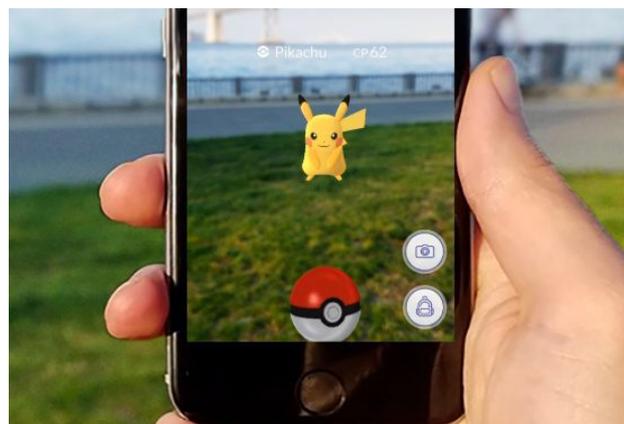


Figura 3.4: Entità AR in Pokémon Go.

### 3.2.2 See-through AR displays

La seconda classe di display utilizzata per le esperienze AR consente all'utente di vedere il mondo “direttamente”, nel senso che vede il mondo reale dalla propria prospettiva (al contrario dei monitor-based in cui il display mostra le immagini aumentate da una telecamera lontana).

Questa tipologia di display è rappresentata dai visori see-through, come gli Head Mounted Display (HMD), che consentono la visione dell'ambiente circostante e allo stesso tempo permettono di percepire l'ambiente di realtà aumentata permettendo di avere le mani libere per agire sulla scena.

Questi sono indossabili dall'utente e gli permettono di vedere il mondo reale attraverso un sistema di telecamere collegato al dispositivo che mira a riprodurre l'effettivo punto di vista degli occhi dell'utilizzatore.

Questi display si distinguono in:

- **optical see-through:** capaci di mostrare grafici, scritte e immagini di sintesi in sovrapposizione alla normale visione dell'utente, arricchendola di utili informazioni. Un divisore di fascio ottico, consistente in uno specchio traslucido, trasmette la luce in una direzione e contemporaneamente la riflette nell'altra. Si tratta quindi di tecnologie parzialmente trasmettenti poiché, guardando attraverso la lente si può vedere l'immagine virtuale sovrapposta a quella reale. Tali divisori di fascio sono molto simili agli Head-Up Display usati dai piloti degli aerei militari. (Figura 3.5)

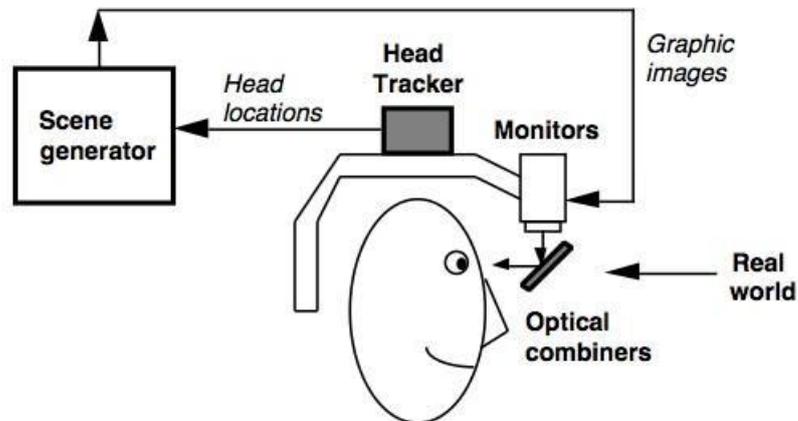


Figura 3.5: Principio di funzionamento basato su optical see-through.

- **video see-through:** usano invece due telecamere, una per ciascun occhio, che acquisiscono l'immagine reale; si tratta, quindi, di un sistema computerizzato che fonde le immagini reali con quelle di sintesi e le invia agli occhi tramite due display. Questa scelta permette di realizzare effetti visivi più complessi, ma ha un fattore diverso dal visore optical see-through,

### CAPITOLO 3. AUGMENTED REALITY

---

perché impone un piano di messa a fuoco costante per tutta la scena e questo rende il sistema poco confortevole. L'immagine del mondo reale è di conseguenza mescolata elettronicamente con l'immagine generata dal computer ed esposta sul display a cristalli liquidi dello schermo dell'HMD. (Figura 3.6)

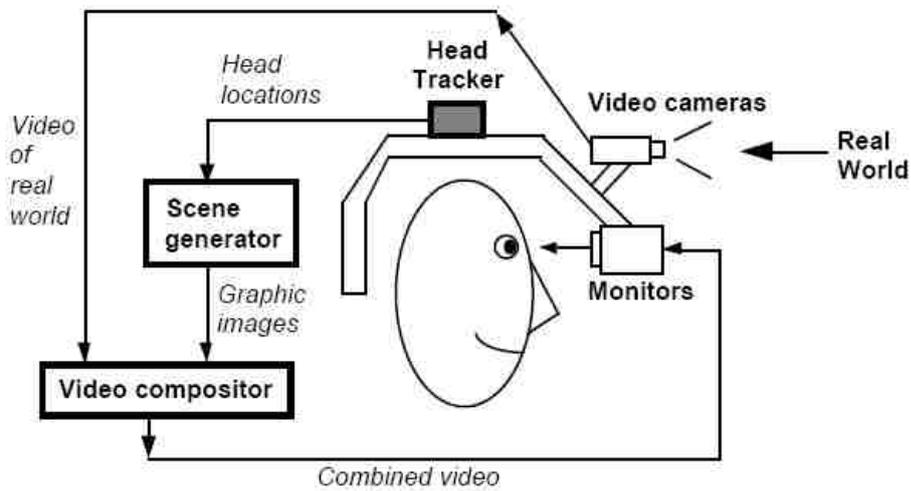


Figura 3.6: Principio di funzionamento del video see-through display.

Un esempio corrente di un dispositivo che rientra nella categoria dei see-through display è Microsoft HoloLens. La Figura 3.7 presenta l'idea che sta alla base di questi dispositivi: ancorare ologrammi nel mondo reale.



Figura 3.7: Esempio di utilizzo dei Microsoft HoloLens: progettare stanze di lavoro con ologrammi integrati nel mondo reale.

In modo analogo ai visual display, che permettono di mostrare elementi virtuali attraverso l'uso della vista, gli audio display producono contenuti audio che possono essere percepiti dall'utente attraverso l'udito.

E' necessario sottolineare che a seconda del tipo di see-through utilizzato possono sorgere problematiche diverse, come sottolineano Fuchs e Ackerman [21]. Un optical see-through HMD deve adeguare rapidamente la posizione e orientamento della testa dell'utilizzatore, e l'intensità della luce per gestire correttamente l'occlusione. Dall'altra parte, un video see-through HMD dovrà assicurarsi che il campo di visualizzazione sia accettabile e, ancora più importante, che il video stesso abbia una frequenza dei fotogrammi stabile. Infatti, un display video see-through mentre si sta assicurando che il mondo reale e il mondo aumentato siano sincronizzati, potrebbe introdurre un ritardo tra i movimenti della testa dell'utente e quello che vede. Questi dettagli vanno gestiti con grande cautela e attenzione, altrimenti gli utenti potrebbero soffrire di nausea.

Un altro elemento che vale la pena ricordare è "l'età" di questi dispositivi. Infatti, sembra che i sistemi HMD siano stati sviluppati già nel tardo 1960, da Sutherland [22] e, nello stesso momento, da Furness [23] per la Force Air americana, con i progetti che portarono poi al programma "Super Cockpit" [24, 25] nel 1986. Di seguito degli esempi di dispositivi HMD:



Figura 3.8: Daqri smart-helmets



Figura 3.9: Epson Moverio BT-200 smart-glasses

### 3.3 Tracciamento del movimento e relative tecniche di computer vision

VR e AR hanno dei requisiti simili in termini di movimento e posizionamento, quindi vengono utilizzate tecniche comuni per soddisfare tali esigenze.

Le applicazioni AR, inoltre, devono spesso riconoscere i punti di riferimento (landmarks) nell'ambiente per ottenere un senso dello spazio in cui si trova l'utente. Questa tecnologia si basa sull'elaborazione di immagini e spesso utilizza dei marcatori (markers) di carta o elementi/caratteristiche rilevate nell'ambiente che diventano i riferimenti spaziali per le entità del mondo aumentato. Il tipo di monitoraggio utilizzato è chiamato *vision-based*, ed è un tipo di monitoraggio ottico (optical tracking), per questo motivo, prima di parlare delle tecniche di tracking visuali proprie dei sistemi AR, si presenta brevemente il concetto di *optical tracking*.

#### 3.3.1 Optical tracking

Le soluzioni ottiche sono probabilmente i sistemi di monitoraggio più ampiamente utilizzati. Il principio generale è che un certo tipo di sensore (tipicamente una fotocamera) traccia le caratteristiche e i modelli dell'ambiente e utilizza dei marcatori attivi/passivi per determinare la posizione e la rotazione dell'oggetto. Le telecamere possono essere "esterne" all'oggetto tracciato (outside-in tracking) o collegate ad esso (inside-out tracking).

#### Marcatori attivi e passivi

Le soluzioni che utilizzano marcatori attivi si basano su emettitori di luce come i LED o le semplici lampadine che vengono posizionate sull'oggetto tracciato. Queste luci sono spesso invisibili agli esseri umani (luci ad infrarossi visibili solo ai sensori a raggi infrarossi) ma necessitano di una fonte di alimentazione. A seconda di come e cosa bisogna tenere traccia, la necessità di batterie può diventare un problema.

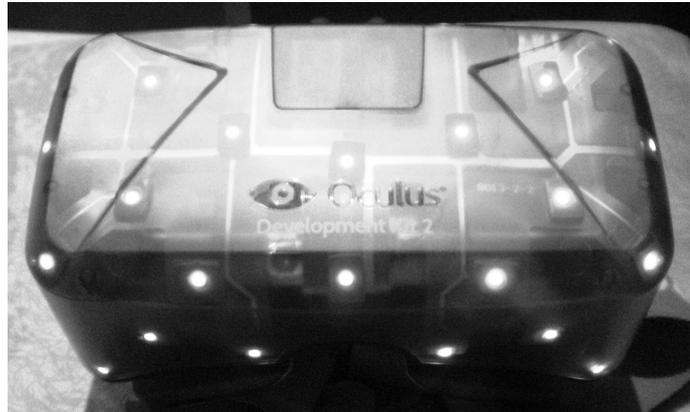


Figura 3.10: Marcatori attivi nell'Oculus Rift DK2

I marcatori passivi non sono luminescenti ma spesso riflettono la luce emessa da una sorgente. La luce emessa e la sua riflessione tipicamente sono invisibili anche per l'occhio umano. I marcatori "carta" sono classificati come marcatori passivi. Come precedentemente menzionato, è anche possibile creare marcatori associandoli a caratteristiche dell'ambiente ad un modello proiettato.

### **Invisible light**

L'utilizzo di sorgenti luminose porta, in genere, i risultati migliori, ma è auspicabile non distrarre l'utente con loro. Le luci ad infrarossi sono "troppo rosse" per essere viste dall'occhio umano e quindi sono quelle utilizzate maggiormente per le invisible light, luci invisibili.

Occorre tuttavia ricordare che l'utilizzo di più sensori a raggi infrarossi può essere un problema (ad esempio se ci fosse una sovrapposizione dei modelli proiettati, i sensori potrebbero confondersi).

Alcuni esempi di tecniche che utilizzano le luci ad infrarossi sono riportate qui di seguito. La prima versione della fotocamera Kinect di Microsofts utilizza una tecnica chiamata "luce strutturata", che prevede essenzialmente modelli ad infrarossi per ottenere informazioni di profondità. Con questi dati è possibile calcolare una rappresentazione 3D della scena che si vede.

La Figura 3.11 mostra il principio del metodo: il modello proiettato viene distorto in base alla sua distanza dalla sorgente.

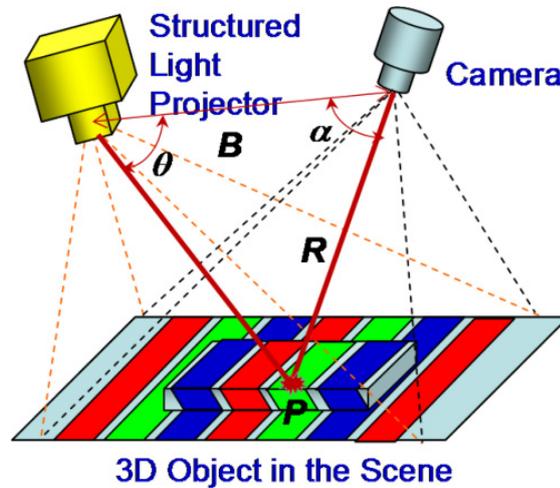


Figura 3.11: Il principio della luce strutturata (*structured light*) nel Kinect v.

Rilasciata alla fine del 2013, la seconda generazione, "Kinect per Xbox One" conosciuta anche come Kinect v2, impiega una tecnologia leggermente diversa per ottenere dati di profondità: il tempo delle misure di flusso (ToF). La luce emessa dal sensore viene ripresa su molte tipologie di superfici e viene misurato il tempo necessario per tornare al sensore. Poiché la velocità della luce è un valore ben noto e costante, è possibile calcolare la distanza dal sensore. Il principio è mostrato nella Figura 3.12, in cui si può vedere che il tempo è effettivamente misurato usando gli spostamenti di fase.

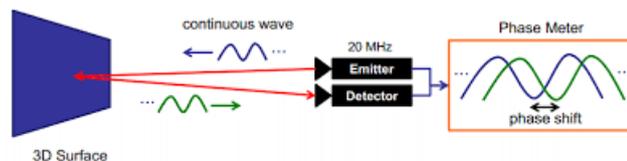


Figura 3.12: The ToF continuous wave technology used in Kinect v2

### 3.3.2 Vision-based tracking

#### Markers

Il tradizionale monitoraggio vision-based utilizza dei modelli di marcatori, generalmente quadrati, in quanto sono considerati la scelta migliore per posi-

zionare i contenuti aumentati, per quanto riguarda i criteri definiti da Owen, Xiao e Middlin [27]. Alcuni esempi di marcatori quadrati sono mostrati nella Figura 3.13.

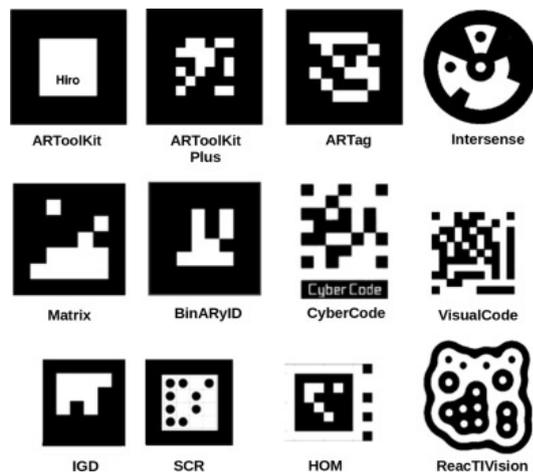


Figura 3.13: Esempi dei markers più conosciuti

ARToolkit è probabilmente il più importante tool open-source di AR. La Figura 3.14 aiuta a capire come il sistema lavora.

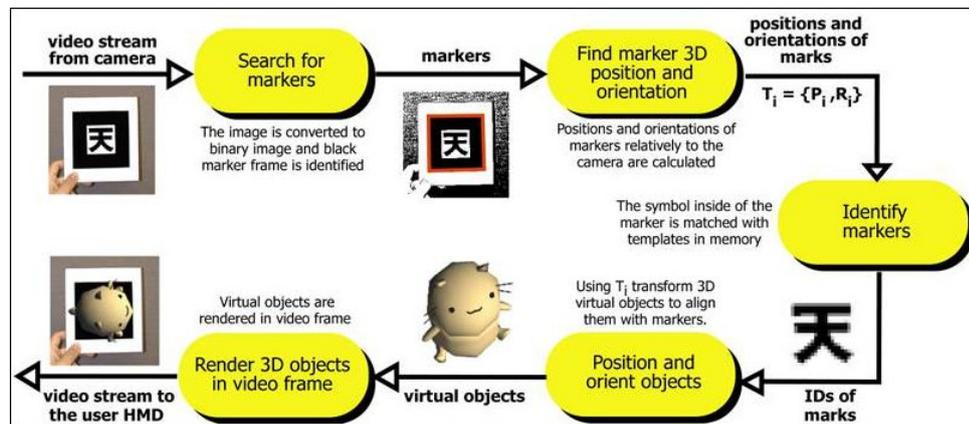


Figura 3.14: Il lavoro di ARToolkit è quello di sovrapporre i contenuti aumentati sui marcatori quadrati

La prima fase ricerca la forma del marker (qualcosa di quadrato e nero). Una volta trovata, viene analizzato il suo contenuto. Se il contenuto incorporato corrisponde a un modello atteso (il marcatore è stato identificato) allora la corrispondente entità aumentata viene sovrapposta ad esso (con la giusta scala, posizione e orientamento, usando gli angoli del marcatore).

### **Figure naturali**

I rilevatori di figure mirano a trovare punti chiave in un'immagine. Queste caratteristiche dovrebbero essere invariante a cambiamenti di posizione, rotazione e scala. Dovrebbero essere robuste all'occlusione, al rumore e al cambiamento della luce, oltre ad essere sufficientemente distinte tra loro. I rilevatori di immagini naturali sono generalmente classificati in tre categorie: rilevatori a singola scala (single-scale), rilevatori a più scale (multi-scale) e i rilevatori a trasformazione affine (affine-invariant).

#### **Single-scale detectors**

I rilevatori a singola scala possono occuparsi di cambiamenti posizionali e rotazionali dell'immagine. Possono anche gestire i cambiamenti di rumore e di illuminazione, ma non sono progettati per affrontare problemi di scalatura. Sono utili quindi per applicazioni AR standard che hanno semplicemente bisogno di un marcatore ma non possono essere utilizzate quando la stessa scena deve essere riconosciuta da punti di vista diversi (che causerebbero cambiamenti di scala). Un esempio tipico è il rilevatore Harris che rileva angoli e bordi guardando i gradienti dell'immagine; misura cioè i cambiamenti di colore o d'intensità nell'immagine).

L'intuizione generale è riportata nella Figura 3.15. Partendo dall'immagine di origine viene assegnato a ciascun pixel un valore che dipende dal gradiente (i risultati vengono visualizzati nella seconda figura della Figura 3.15, dove i valori più elevati sono visualizzati in rosso). Utilizzando una soglia, vengono mantenuti solo punti interessanti (i punti bianchi della terza immagine della stessa figura).

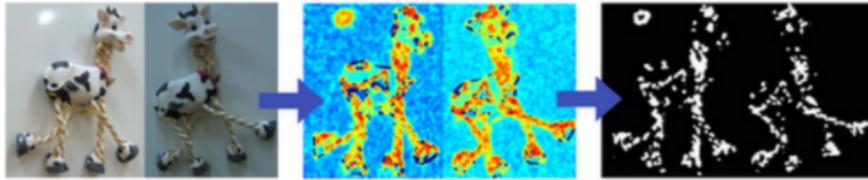


Figura 3.15: Flusso di lavoro del rivelatore di Harris.

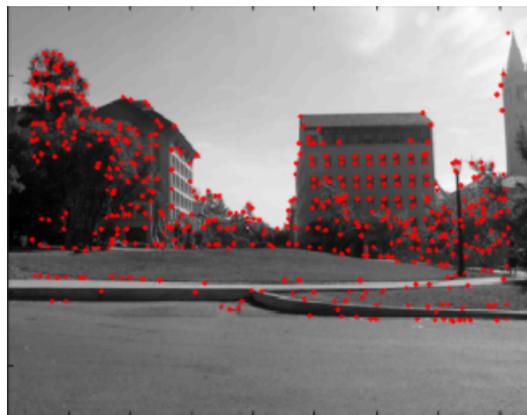


Figura 3.16: Output del rivelatore di Harris su un'immagine. I puntini rossi rappresentano le caratteristiche rilevate.

Infine, vengono mantenuti solo i massimi locali (i valori più alti del "quartiere locale"), questi sono i punti chiave restituiti come output, come mostrato in un altro esempio nella Figura 3.16.

### Multi-scale detectors

Il rivelatore di Harris può essere facilmente adattato per affrontare i cambiamenti di scala (se si conosce la formazione esatta). Purtroppo, negli scenari reali, i cambiamenti di scala sono sconosciuti, quindi bisogna trovare un modo per rilevare i punti interessanti nelle diverse scale di una figura. Identificando le regioni dell'immagine che presentano proprietà (ad esempio luminosità o colore) diverse dall'ambiente circostante, si formano delle "macchie". Per scoprirle, il metodo principale è quello di utilizzare il laplaciano dei filtri gaussiani (LoG). Questi permettono di *lisciare* l'immagine con un filtro gaussiano molto sensibile al rumore, e in pratica, nebulizzano la regione per ottenere le macchie. LoG

può quindi essere applicato per trovare una scala locale ad una specifica regione dell'immagine, il che significa che è selezionata automaticamente la scala giusta per quella regione.

### **Affine-invariant detectors**

I rivelatori a singola scala, discussi in precedenza, mostrano invarianza alle traduzioni e alle rotazioni. Da parte loro, i rivelatori a più scala possono anche gestire le scale uniformi e, in una certa misura, le trasformazioni affini. Quest'ultima è la capacità di gestire la mappatura del taglio e la scalatura non uniforme; con "mappatura del taglio" ci si riferisce essenzialmente alla possibilità di visualizzare la scena da una diversa prospettiva mantenendo il parallelismo, come si vede nella Figura 3.17.

I rilevatori affine-invariant fanno un ulteriore passo in avanti: sono in grado di gestire trasformazioni affini significative. Un esempio di questi rilevatori è quello di Mikolajczyk e Schmid, proposto nel 2004. L'idea di questo metodo è quella di estrarre la forma caratteristica (in contrapposizione alla scala caratteristica vista precedentemente) della figura rilevata. Le macchie a cerchio vengono sostituite da macchie ad ellissi (Figura 3.18) con lunghezze d'asse che dipendono dallo stesso valore definito dal gradiente utilizzato nel detector di Harris che abbiamo discusso prima.



Figura 3.17: Effetti di un affine-invariant detectors; mappatura e parallelismo sono preservati.

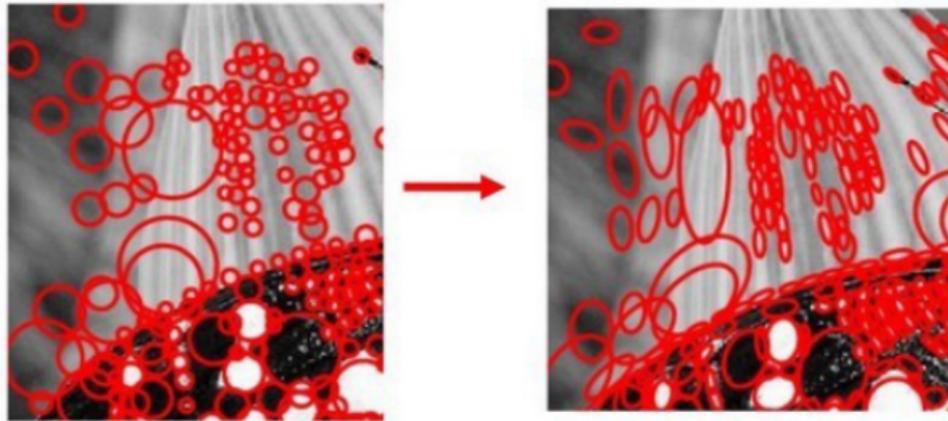


Figura 3.18: Mikolajczyk e Schmid concetto chiave: le forme caratteristiche sono evidenziate da ellissi di dimensioni diverse.

### 3.3.3 SLAM

Il problema di localizzazione e mappatura simultanea (SLAM - simultaneous localization and mapping) descrive la mappatura di un ambiente sconosciuto da parte di un robot mobile. Infatti, senza alcuna conoscenza precedente dell'ambiente, il robot deve risolvere contemporaneamente due problematiche: localizzare sè stesso e mappare l'ambiente.

Il problema SLAM è fortemente legato con la realtà aumentata, specialmente per l'AR *spaziale*. Questo perchè i dispositivi devono avere un senso dell'ambiente in cui si trovano per integrare correttamente le entità aumentate nei loro dintorni. Questa sezione presenta quindi le basi del problema. Anche se molti ricercatori stavano già lavorando alla mappatura e alla localizzazione, la struttura del problema SLAM e l'acronimo sono stati presentati solo nel 1996 da Durrat Whyte, Rye e Nebotin. Essi danno una definizione formale del problema come illustrato nella Figura 3.19 e descritto in seguito. Un robot si muove liberamente in un ambiente mentre utilizza i suoi sensori per tracciare i punti di riferimento sconosciuti.

At time  $k$ , we have:

- $x_k$  the vector describing the position/orientation of the robot
- $u_k$  the vector, applied at time  $k - 1$  to move the vehicle to  $x_k$  at time  $k$
- $m_i$  the vector describing the position of the  $i^{\text{th}}$  stationary landmark
- $z_{ik}$  the observation, taken from the robot, of the  $i^{\text{th}}$  landmark at time  $k$

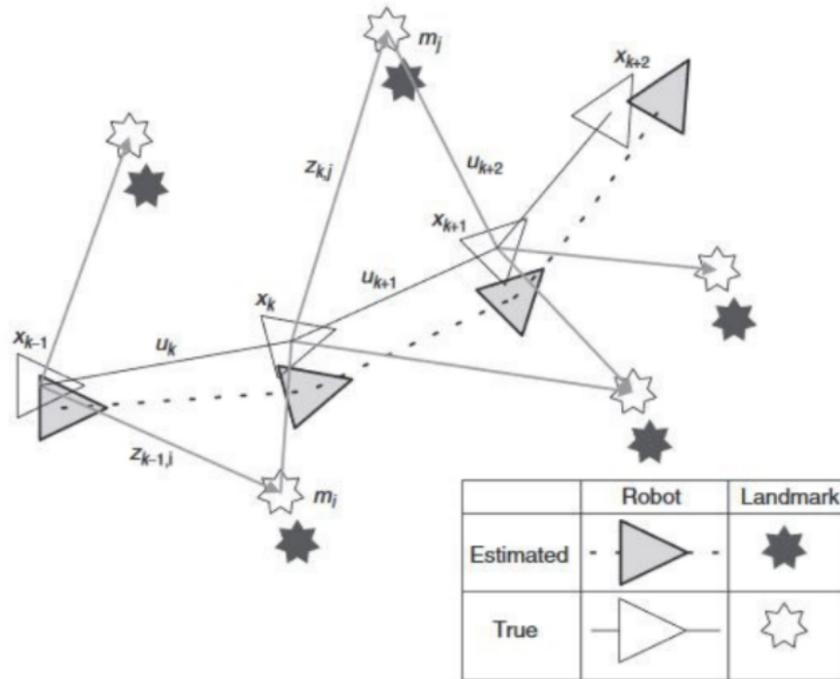


Figura 3.19: Il problema SLAM com'è descritto in [30].

Per estrarre gli elementi caratteristici naturali, si utilizzano gli algoritmi descritti precedentemente. In questo modo è possibile definire la posizione della telecamera e dei punti di riferimento presenti nell'ambiente, ma poichè il rumore e le imprecisioni sono inevitabili, sono stati sviluppati dei metodi per cercare di limitarli. Un'opzione ampiamente utilizzata è l'*Extended Kalman Filter* (EKM) [28] unito, a volte, con reti neurali appositamente addestrate [29].

### 3.4 Piattaforme per Augmented Reality

Di seguito si vogliono elencare alcuni dei frameworks più utilizzati per lo sviluppo di applicazioni AR.

- **Vuforia**<sup>1</sup> è una delle piattaforme più popolari per la realizzazione di applicazioni di realtà aumentata. Fornisce supporti per Android, iOS, UWP (Universal Windows Platform) e Unity Editor. Tra le funzionalità principali del software ricordiamo: il riconoscimento di tipi diversi di oggetti visivi (caselle, cilindri, piani), riconoscimento di testo, ambienti e VuMark (una combinazione di immagini e QR-code). Inoltre, utilizzando il Vuforia Object Scanner, è possibile scannerizzare e creare degli "object targets". Il processo di riconoscimento degli elementi può essere implementato utilizzando un database, locale o sul Cloud. Tutti i plugin e le funzionalità della piattaforma sono gratuite ma includono delle limitazioni; come il numero di VuMark utilizzabili e il numero di riconoscimenti nel Cloud. Il plugin per Unity è semplice da integrare e molto potente.
- **ARtoolKit**<sup>2</sup> è una libreria di monitoraggio open source per sistemi di realtà aumentata, mantenuta su GitHub. E' una piattaforma molto utilizzata; nel 2004 alla sua pubblicazione, ha riscontrato più di 160.000 downloads. Fornisce supporti per Android, iOS, Linux, Windows, Mac OS e Smart Glasses. Utilizza funzionalità di monitoraggio video che calcolano la reale posizione e orientamento della telecamera rispetto ai marker quadrati o rispetto ai marker naturali, in tempo reale. Conosciuta la posizione della telecamera reale, inserisce una telecamera virtuale che permette di visualizzare modelli di computer grafica 3D, esattamente sopra il marker scelto.
- **Wikitude**<sup>3</sup> è un fornitore di tecnologia AR per dispositivi mobili con sede a Salisburgo, in Austria. E' stato fondato nel 2008 e inizialmente permetteva lo sviluppo di applicazioni di realtà aumentata attraverso l'app Wikitude World Browser. Nel 2012, la compagnia ha ristrutturato la sua proposta lanciando il Wikitude SDK; un framework di sviluppo che utilizza il riconoscimento e il monitoraggio di immagini, e tecnologie di geolocalizzazione. Fornisce supporti per lo sviluppo su piattaforme native e

---

<sup>1</sup><https://www.vuforia.com/>

<sup>2</sup><https://github.com/artoolkit>

<sup>3</sup><https://www.wikitude.com/>

javascript, iOS e Android, con estensioni per Unity, Cordova, Titanium e Xamarin.

- **Kudan**<sup>4</sup> secondo le recensioni è il principale rivale di Vuforia. Anch'esso rende lo sviluppo di sistemi di realtà aumentata molto semplice. Fornisce supporti per piattaforme Android e iOS. Utilizzando la tecnologia SLAM permette il riconoscimento di immagini e di oggetti 3D, fornisce una facile generazione del database nello Unity Editor. Sempre secondo le recensioni, però, le app realizzate in Kudan si bloccano facilmente sui dispositivi.
- **Maxst**<sup>5</sup> è disponibile per diverse piattaforme, quali, Android, iOS, Mac OS e Windows. Offre strumenti per il riconoscimento di immagini e per la generazione di database online tramite il Tracking Manager. Per scansionare gli oggetti 3D vengono utilizzate delle applicazioni Android e iOS. In Unity Editor funziona con la versione a 32bit.

---

<sup>4</sup><https://www.kudan.eu/>

<sup>5</sup><https://developer.maxst.com/>

## Capitolo 4

# Web of Augmented Things

Avendo letto i precedenti capitoli ci siamo fatti un'idea tecnica di quelle che sono le *augmentation technologies* e dell'impressionante sviluppo che hanno mostrato in questi anni. Possiamo dire che con *augmentation technologies* si intendono tutte quelle tecnologie che arricchiscono/aumentano il mondo fisico e la realtà con funzionalità e servizi digitali computazionali. In questa prospettiva, i *pervasive/ubiquitous computing system* sono tecnologie di *augmentation*, e ulteriori esempi sono dati dalla realtà aumentata (AR) e dalla realtà mista (MR).

La realtà aumentata può essere definita come il mezzo con cui delle informazioni digitali vengono aggiunte (sovrapposte) al mondo fisico e rese visibili ad un utente in modo dipendente dalla sua posizione e dalla sua prospettiva. Tecnologie commerciali come Microsoft HoloLens<sup>1</sup> o Meta<sup>2</sup> permettono di aumentare gli ambienti fisici interni con ologrammi piuttosto sofisticati, con i quali gli utenti possono interagire utilizzando gli *smart glasses*. Gli “occhiali intelligenti” sono il principale esempio di dispositivi di calcolo *wearable* e possono essere incorniciati come tecnologie di *human augmentation*. Il punto chiave di queste tecnologie è che consentono alle persone di poter svolgere attività lavorative e operazioni fisiche in condizioni *hands-free*, ovvero senza l'utilizzo di dispositivi come smartphone e tablet che obbligano l'impiego delle mani. Permettono quindi di percepire e interagire con il sistema senza fermare o distrarre l'utente da ciò che sta facendo con il proprio corpo. Concettualmente, questi dispositivi danno la possibilità di estendere le capacità cognitive delle persone, migliorando il modo in cui svolgono i loro compiti [31].

---

<sup>1</sup>[www.microsoft.com/microsoft-hololens/en-us](http://www.microsoft.com/microsoft-hololens/en-us)

<sup>2</sup>[www.metavision.com/](http://www.metavision.com/)

L'integrazione di tutte le tecnologie di *augmentation*, *pervasive-computing*, *wearable* e *augmented reality*, consente di concepire nuovi tipi di ambienti intelligenti che permettono di ripensare al modo in cui le persone lavorano, interagiscono e collaborano. Ci riferiamo a tali sistemi come *augmented world* (AWs) [15]. Considerando il progresso delle tecnologie hardware, in breve tempo il grado di sofisticazione fornito da un AW dipenderà in gran parte dal software che dovrà sfruttare al meglio le tecnologie abilitanti. Sul versante nel *pervasive computing*, la proposta attuale è quella di Web of Things (WoT), suggerendo l'utilizzo del web e dei relativi protocolli come base per la creazione di applicazioni che consentono lo sviluppo dello stack di Internet of Things (IoT).

L'idea fondamentale discussa in questo capitolo è quindi quella di applicare i principi e i protocolli di WoT anche agli *augmented world*, definendo ciò che qui chiamiamo **Web of Augmented Things** (WoAT) [2]. A questo proposito, WoAT prevede lo sviluppo di sistemi di realtà aumentata composti da ologrammi visti come Augmented Things, ognuno con un URL e un'interfaccia web RESTful API, che da la possibilità di interagire con esse usando le tecnologie web standard.

Il WoAT può essere considerato un'estensione conservativa (una specializzazione) del WoT, in modo tale che gli oggetti intelligenti/*things* di un sistema WoT possano essere accoppiate ad un certo ologramma, arricchendo ulteriormente le loro funzionalità. Nella prima sezione si evidenziano gli aspetti salienti di AW. Nella seconda sezione si introduce nel dettaglio l'idea di WoAT, e nella terza si propone il modello e gli aspetti fondamentali di un'architettura che supporta i concetti di Web of Augmented Things.

## 4.1 Augmented Worlds - concetti chiave

Introduciamo il concetto di Augmented World (AW) per catturare in un unico modello uniforme gli aspetti principali delle *augmentation technologies* e della loro integrazione/convergenza. In termini generali, un AW può essere definito come un'applicazione software che arricchisce le funzionalità di un particolare ambiente fisico (ad esempio, una stanza, un edificio, una città, ecc.) mediante oggetti computazionali a pieno titolo situati nello spazio, che gli utenti e gli agenti software<sup>3</sup> possono percepire e con i quali possono interagire (vedere la

---

<sup>3</sup>Con agenti software ci si riferisce genericamente a qualsiasi tipo di componente software autonomo.

figura 1). Ci riferiamo a questi oggetti computazionali come entità aumentate (augmented entity - AE).

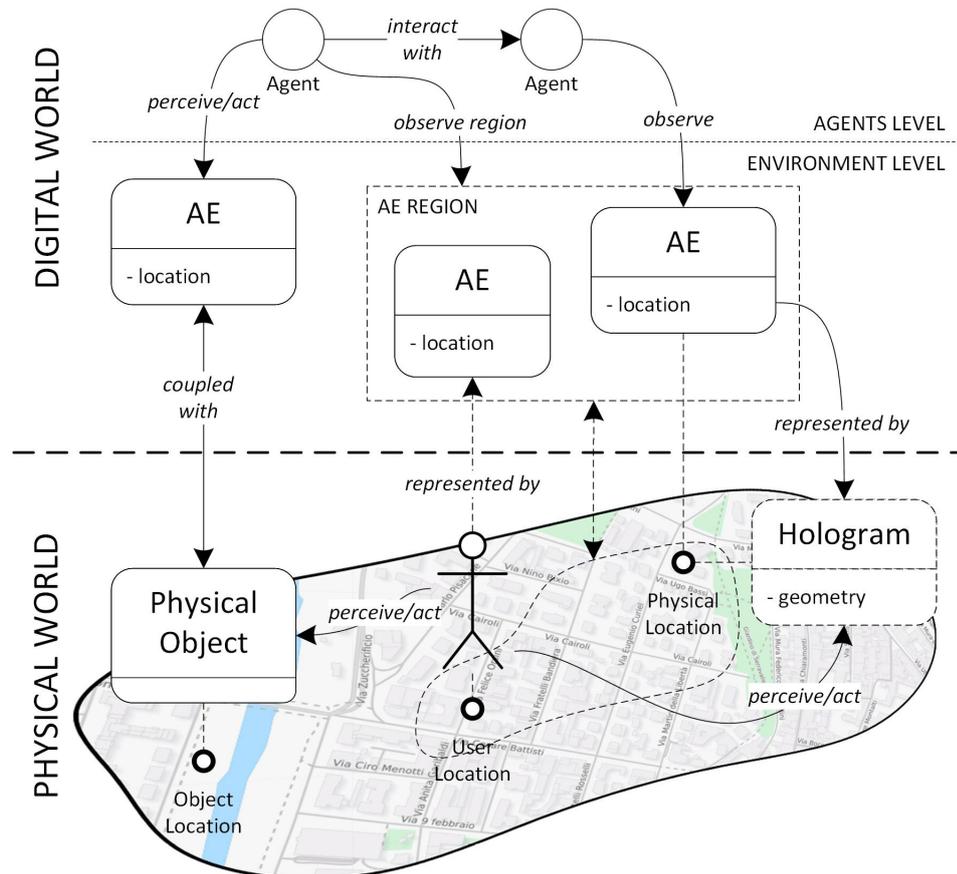


Figura 4.1: Rappresentazione di un Augmented World.

Un AE, come oggetto computazionale, fornisce un'interfaccia con azioni che gli agenti possono sfruttare per agire su di essa, modificandone lo stato. Inoltre, un AE fornisce un insieme di proprietà osservabili, che rappresentano informazioni sullo stato dell'entità che può essere percepito e osservato dagli agenti. Le proprietà osservabili possono essere utilizzate per abilitare le forme di interazione *event-driven publish/subscribe*, in modo che le modifiche alle proprietà osservabili vengano automaticamente notificate agli agenti interessati (che han-

no deciso dinamicamente di seguire l'entità).

Questo modello è intenzionalmente ispirato al modello concettuale A&A, introdotto nel contesto dei sistemi ad agenti [32].

Un AW e AE sono caratterizzati da cinque funzionalità principali che discuteremo nel resto di questa sessione.

### 4.1.1 Accoppiamento spaziale

Come abbiamo già detto, gli AE hanno una posizione specifica nella realtà fisica. Un'entità aumentata può essere creata o istanziata in qualsiasi punto di una regione coperta dall'AW, specificando la sua posizione (una posizione geografica o un riferimento ad un oggetto reale/fisico) rispetto al sistema di riferimento locale (comunque definito all'interno del mondo aumentato). Questa proprietà viene specificata come *spatial coupling*.

Un AE può essere un oggetto computazionale o un'entità software, che incapsula uno stato e un comportamento preciso. In questo senso, un mondo aumentato diventa un contesto che propone e offre servizi e/o funzionalità complesse, in cui le entità software potrebbero diventare utili per esprimere una sorta di arricchimento per gli oggetti e gli abitanti del mondo fisico.

### 4.1.2 Ologrammi

Un'entità aumentata, nella visione di AW, non è "aumentata" nel senso di realtà aumentata. Infatti, un AE esiste in un AW anche se non è percepibile in termini d'impatto visivo. In altre parole, un AE potrebbe essere rappresentata solo come un oggetto computazionale (una parte di un software) che esegue da qualche parte i suoi compiti computazionali. Nonostante questo, gli *augmented worlds* danno la possibilità di associare ad un'entità aumentata una geometria che consente di rappresentarla nel mondo reale.

Se un'entità aumentata deve essere percepibile dagli esseri umani attraverso dispositivi specifici (come gli *smart glasses*) è possibile associarle uno o più ologrammi, ciascuno dei quali è descritto come un insieme di forme 2D/3D definite in relazione ad un sistema di riferimento locale ed è considerato come parte dello stato interno dell'entità aumentata.

### 4.1.3 Scoperta e osservazione

La spazialità delle entità aumentate può essere sfruttata per arricchire i modi in cui questi oggetti computazionali vengono scoperti, osservati e più in generale come interagiscono tra di loro o come vengono utilizzati/percepiti dagli agenti. In particolare, gli agenti che vivono in un AW possono sfruttare adeguate funzioni di ricerca spaziale per scoprire l'identificatore o il riferimento di tutte le entità aumentate disponibili in alcune regioni, ad esempio, nella regione in cui si trova l'agente stesso.

Inoltre, un AW consente agli agenti di monitorare/osservare in continuazione una regione in uno stile più asincrono/orientato agli eventi, per essere notificati quando una nuova AE entra o esce dalla regione.

### 4.1.4 Embedding fisico

La nozione *augmentation* in AW significa anche estendere le funzionalità di oggetti fisici esistenti. Tale estensione può essere utile a due livelli diversi (ma connessi). Il primo è per gli utenti che interagiscono con gli oggetti fisici, estendendo le dotazioni degli oggetti tramite interfacce virtuali - descritte come geometria di un ologramma correlato - per ispezionare lo stato dell'oggetto fisico. Il secondo è quello di arricchire le funzionalità dell'oggetto fisico, sfruttando le capacità di calcolo fornite dal *augmented layer*.

Questo tipo di "aumento" richiede in genere un *accoppiamento* tra gli oggetti fisici e le corrispondenti entità aumentate, in modo che le modifiche allo stato fisico degli oggetti siano monitorate e notificate nel livello aumentato e viceversa. Quando questo non è possibile, ad esempio a causa di assenza di connettività, l'infrastruttura dell'*augmented world* deve aggiornare una sorta di meta-informazione per ogni informazione/dato legato al mondo fisico (come "grado di freschezza"). Questo serve per determinare se e quando una specifica informazione/dato non può più allinearsi con lo stato reale.

### 4.1.5 Utente

L'interazione con gli utenti è uno degli aspetti principali degli AW. Infatti l'obiettivo principale di questi mondi è quello di modellare applicazioni in cui gli utenti umani sono impegnati in attività di qualche tipo in un certo ambiente fisico.

Un mondo aumentato è in genere un'applicazione multiutente, in cui più utenti

percepiscono, condividono e interagiscono con le stesse entità aumentate. Un modo naturale per modellare un utente in un mondo aumentato è associarlo ad un'entità aumentata - come una sorta di corpo amplificato - con funzionalità utili per il controllo dell'ingresso/uscita dell'utente e uno stato computazionale che reifica le informazioni dinamiche che possono essere rilevanti a livello di applicazione. Ad esempio, un agente personale dell'utente può essere a conoscenza dell'insieme di entità aumentate che si trovano vicino all'utente seguendo/osservando una regione intorno al corpo aumentato. Viceversa, altri agenti (che controllano, ad esempio, altri ologrammi che vivono nel AW) possono essere consapevoli della posizione dell'utente e del suo stato, monitorando/osservando il suo corpo aumentato.

## 4.2 WoT + AW: Web of Augmented Things

Web of Things[9] ha introdotto l'idea fondamentale di definire un livello di applicazione comune per IoT basato sul web e sui relativi protocolli e strumenti. Ciò consente di ottenere notevoli vantaggi in termini di sviluppo del software e di ingegneria, in particolare per quei sistemi complessi che potrebbero avere bisogno di affrontare problematiche con tecnologie e protocolli eterogenei. Oltre all'utilizzo del web come protocollo a livello applicazione, l'architettura Web of Things che si propone in [9] identifica una pila di strati che affrontano aspetti diversi di tali sistemi: dal consentire l'accesso alle *things* fino a modificarle/crearle (come si può vedere dalla Figura 4.2).

L'idea fondamentale di questo articolo è quella di sfruttare gli stessi principi e lo stesso stack per gli AW e, più in generale, per tutti i sistemi basati su AR/ologrammi, quindi, in pratica, per integrare nel WoT anche i mondi aumentati e la realtà aumentata.

Ciò consente di concepire l'ingegnerizzazione di AW aperti e complessi, composti da entità aumentate eventualmente sviluppate utilizzando diverse piattaforme e che possono essere percepite usando diverse tecnologie AR.

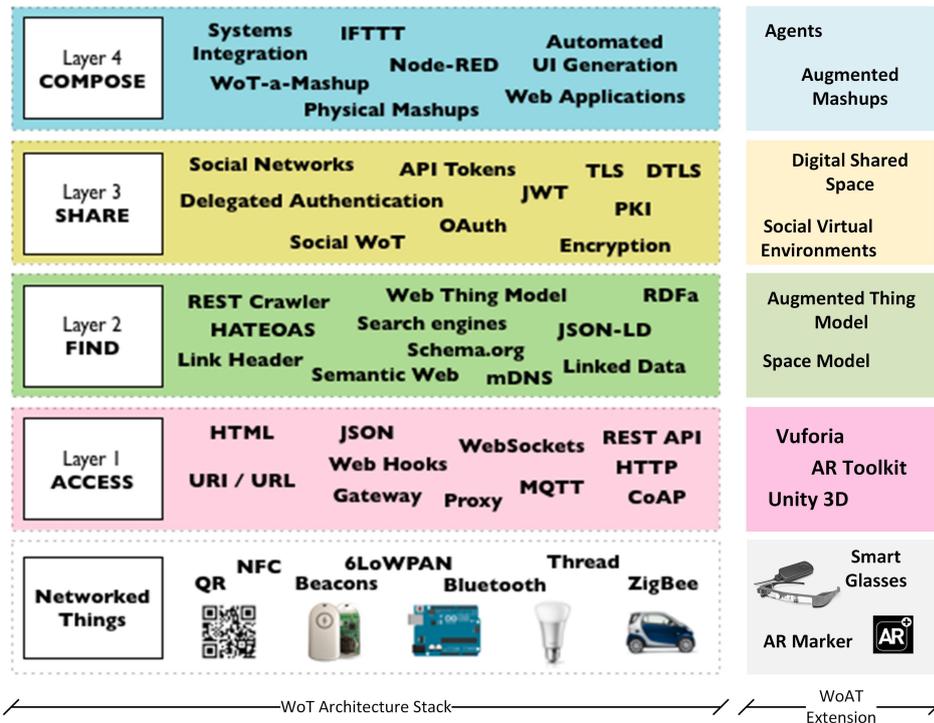


Figura 4.2: Specializzazione del WoT con AW.

Alcune *augmented entity* potrebbero essere totalmente virtuali (mentre si trovano in una certa posizione dello spazio fisico), altre possono essere associate ad alcuni oggetti fisici. Tali oggetti potrebbero essere semplici "cose" fisiche, o *things* di WoT: in questa prospettiva, WoAT riguarda i sistemi WoT aumentati da uno strato di AR.

Il nucleo di WoAT si basa sul concetto di vedere una *Augmented Thing* (AT) come una *things* nella prospettiva WoT che rappresenta un'entità aumentata che vive in un AW.

Come ogni altra "cosa" in WoT, una AT:

- Possiede un URL di origine, che corrisponde al suo indirizzo di rete;
- E' dotata di un'interfaccia web REST-ful, basata su messaggi auto-descrittivi, operazioni HTTP (GET, POST, PUT, DELETE, HEAD) e meccanismi orientati all'evento (ad es. web socket).

Il modello delle *things* aumentate (AT) può essere considerato una specializzazione del modello delle risorse/dati di WoT, come definito in [9] (mostrato in Figura 4.3), rispettando i principi di AW e in particolare il modello di Augmented Entity. Le azioni (operazioni) e le *proprietà osservabili*, che sono il nucleo del modello dell'entità aumentata, vengono naturalmente mappate/rappresentate da azioni e proprietà come definite nel modello delle *things* di WoT.

In questo:

- **GET** può essere utilizzato per recuperare le proprietà (osservabili) di una Augmented Things (cioè, un'entità aumentata).
- **POST** può essere utilizzato per inviare azioni ad un AT, come per le *things* di WoT.

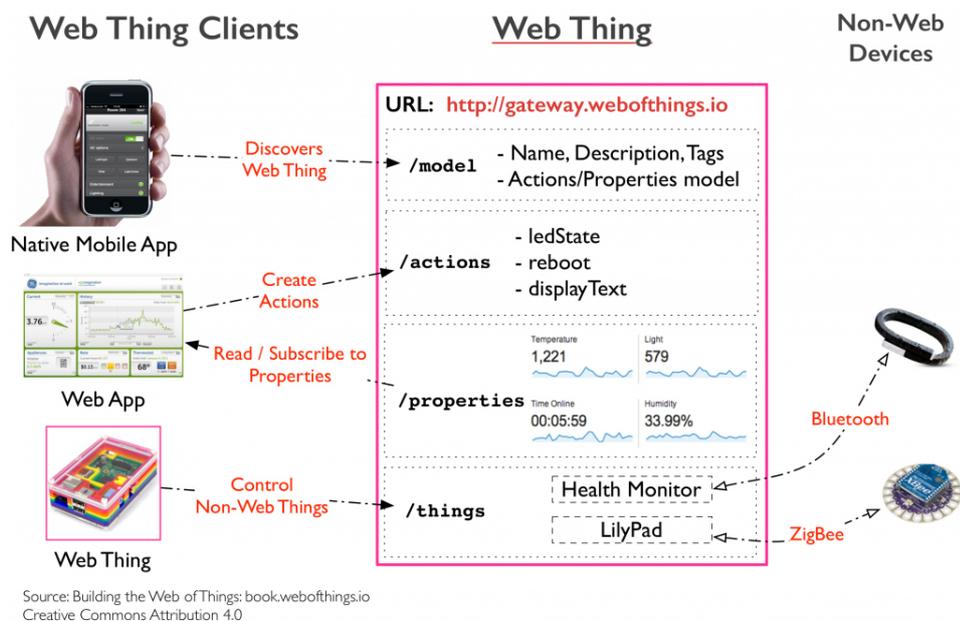


Figura 4.3: Modello di WoT.

Inoltre, il modello di una AT include funzionalità/proprietà specifiche, provenienti dal modello concettuale di Augmented World definito precedentemente. Un primo insieme di queste specifiche è legato al suo *accoppiamento con lo spazio fisico*:

- Ogni AT ha una posizione nel mondo/spazio fisico definito nel sistema di riferimento dentro alla regione che caratterizza l'AW. Questo è rappresentato da una proprietà specifica:

`/properties/location`

La posizione è importante anche nel caso di sistemi AR basati su marcatori in quanto il marker viene sfruttato per definire il sistema di riferimento dove l'entità aumentata è immersa.

- La posizione può includere informazioni sia sulla locazione sia sull'orientamento dell'entità. Oltre a una posizione, ogni AT ha un'*estensione nello spazio* (che può anche essere un punto singolo), rappresentata da una proprietà specifica:

`/properties/extension`

Poi, seguendo il modello concettuale di AW, una AT può avere uno o più ologrammi che definiscono come l'entità viene percepita dagli utenti umani e come possono interagire con essa. Queste informazioni sono rappresentate da risorse:

`/properties/holograms`

Infine, in questo contesto, le risorse/things definite nel modello WoT possono essere riutilizzate per specificare informazioni sull'*embedding fisico*, riferendosi a *things* accoppiate con le AT.

Oltre alle "cose aumentate", anche lo stesso *augmented world* può essere modellato (opportunamente) come una risorsa di WoT. Una proprietà dovrà riguardare il *set dinamico di AT* appartenenti al mondo aumentato, quindi per tenerne traccia e renderlo osservabile. Mentre le azioni dovranno includere come principali primitive quelle che *creano/rimuovono* una AT, quelle che *cercano* le AT appartenenti ad una specifica regione all'interno dell'AW e quelle che permettono di *iniziare/smettere di osservare* una regione specifica dell'AW.

### 4.3 Infrastruttura per WoAT

In questa sezione tracciamo gli elementi principali di una possibile infrastruttura per WoAT, da un punto di vista progettuale e architettonico, così come definita in [2].

Fondamentalmente, possiamo considerarla come un'estensione di WoT, arricchito con:

- Supporti infrastrutturali specifici per l'implementazione delle specifiche astrazioni e funzionalità del WoAT.
- Librerie sul lato client che consentono l'esperienza basata su AR per gli utenti umani.

Un modello astratto dell'infrastruttura proposta è rappresentato dalla Figura 4.4. Ogni parte dell'infrastruttura, a prescindere dal fatto che sia dedicata a sostenere l'Infrastructure Node, piuttosto che gli esseri umani, o gli agenti/componenti software o le *things* fisiche, è logicamente costruita sullo strato WoT, che si trova a sua volta sopra lo strato di IoT in modo che l'infrastruttura sia progettata per supportare anche le *things* fisiche.

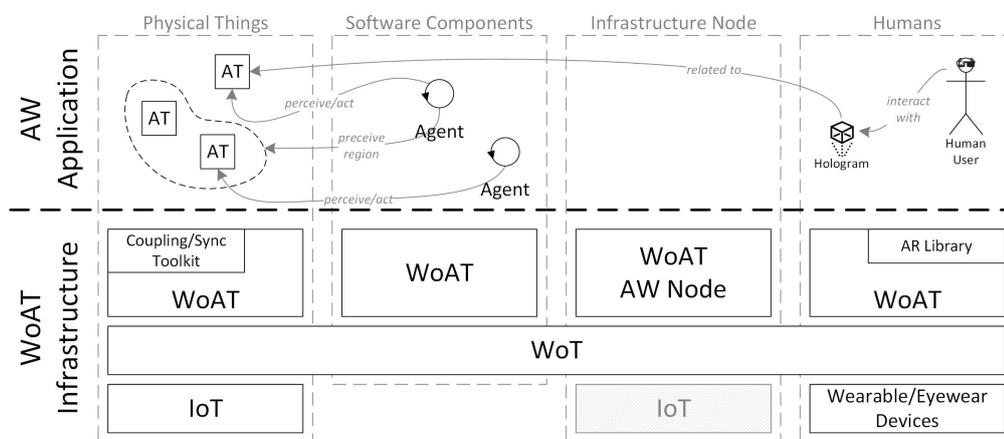


Figura 4.4: Infrastruttura WoAT.

Potenzialmente, questo potrebbe anche essere vero per la parte dell'infrastruttura che riguarda l'Infrastructure Node, a seconda della sua specifica distribuzione, che può essere compresa tra i dispositivi *embedded* e il *cloud computing*. Indipendentemente dalla parte specifica dell'infrastruttura, i *layer* WoAT

forniscono astrazioni per supportare le funzionalità specifiche delle *things* aumentate, come spiegato nella sezione precedente.

L'Infrastructure Node ospita l'AW, fornendo sostanzialmente il relativo sistema di riferimento del mondo digitale in relazione a quello fisico, e le funzioni di ricerca per recuperare entità e regioni aumentate.

Le entità aumentate o meglio, *things* aumentate (nella prospettiva di WoAT), utilizzano questa parte dell'infrastruttura per registrarsi ed esporre proprietà particolari e l'elenco di funzionalità utilizzabili per poter agire su di esse.

Inoltre, l'Infrastructure Node potrebbe essere utilizzato anche per ospitare alcune "cose" aumentate relative a certe posizioni fisiche, ma che non hanno una contro parte fisica. La parte dell'infrastruttura dedicata al supporto delle "cose" fisiche, aggiunge al WoAT delle funzionalità specifiche per mantenere l'accoppiamento tra le "cose" reali e le rispettive "cose" aumentate; aggiornando ed eseguendo una sincronizzazione degli stati dal mondo fisico al mondo digitale e viceversa.

Inoltre, una parte specifica dell'infrastruttura è dedicata a supportare componenti software autonomi, ad es. agenti. L'infrastruttura propone l'astrazione per garantire agli agenti la capacità di unirsi agli AW, così da interagire con le *augmented things*.

Infine, l'infrastruttura aggiunge anche il supporto per le librerie di AR, per gestire la visualizzazione e l'interazione con gli ologrammi da parte degli utenti dotati di dispositivi appropriati (ad es. *smart glasses*).



# Capitolo 5

## Sperimentazione dell'idea di WoAT

### 5.1 Definizione del problema

Il progetto di tesi è nato per sperimentare e verificare la possibilità di utilizzo dei concetti di WoAT in un'applicazione pratica. Allo scopo è stato scelto di realizzare un programma che utilizza le tecnologie di realtà aumentata per estendere le funzionalità di *things* fisiche. Per la parte WoT è stato sfruttato un progetto parallelo realizzato da miei colleghi; con questa simbiosi abbiamo testato un esempio di WoAT.

L'applicazione permette di visualizzare e interagire con un'entità aumentata, in quest'ottica meglio "*thing* aumentata", la cui manipolazione agisce su una *thing* del mondo reale. Nello specifico il software permetterà di avere l'estensione aumentata di un Led il cui stato potrà essere pilotato anche da due bottoni virtuali. Secondo i principi del WoAT discussi precedentemente, l'oggetto aumentato possiede un URL che lo identifica e un'interfaccia web Rest-full per accedere alle sue proprietà osservabili. Questo ci consente di tenere separati i dati dell'entità (model) dalla sua visualizzazione (view).

Il progetto è stato realizzato in Android Studio utilizzando gli strumenti per la generazione nativa Android, insieme al framework Vuforia per la gestione della realtà aumentata.

## 5.2 Tecnologie

- **Vuforia SDK** come libreria per la gestione della Realtà Aumentata sull'applicazione Android. Usa una tecnologia di Computer Vision per riconoscere e tracciare immagini (image targets) e semplici oggetti 3D. Queste capacità di riconoscere le immagini target ci permette di posizionare e orientare gli oggetti virtuali in relazione al mondo fisico.
- **Php** è un linguaggio di scripting interpretato per applicazioni lato server specialmente indicato per lo sviluppo web. In questo progetto è stato utilizzato per l'implementazione del Server Rest su Apache e l'accesso al database MySql che contiene gli oggetti del mondo virtuale/aumentato.
- **Android SDK** usato per la realizzazione dell'applicazione client su smartphone e *smart glasses* Android. L'Android SDK è un kit di sviluppo software Android che include una serie di strumenti come un debugger, librerie, un emulatore, documentazione e codici esemplificativi che permettono di realizzare applicazioni Android utilizzando il linguaggio Java.

## 5.3 Progettazione del software

La scelta progettuale adottata è quella di avere un database distribuito che contenga la struttura e lo stato degli oggetti aumentati e un'applicazione, lato client, che li visualizza e ci interagisce attraverso chiamate Rest-full API. Questo consente di avere più client che vedono e agiscono sulle stesse *things* aumentate.

Quindi la struttura del sistema è composta da due sottosistemi:

- Server Rest realizzato con Php su Apache.
- Applicazione client realizzata con Android SDK e Vuforia SDK

### 5.3.1 Server Rest

Il server Rest è stato realizzato in Php su macchina remota con sistema operativo Linux, web server Apache e database MySql. Ho utilizzato una libreria php *open source* messa a disposizione da Jac Wright<sup>1</sup> per implementare in modo

---

<sup>1</sup><https://github.com/jacwright/RestServer>

semplice e intuitivo le API Rest. L'oggetto viene gestito da queste API per quanto riguarda la sua rappresentazione, posizione e stato nel mondo virtuale. L'interrogazione delle proprietà osservabili è effettuata tramite metodi GET, mentre la modifica delle proprietà variabili tramite metodi PUT e/o POST. In questo caso, la risorsa controllata è un singolo oggetto chiamato "led"; le sue caratteristiche modificabili sono: la scala, lo stato (on, off), la velocità di rotazione e la sua posizione (a tre valori, per gli assi x,y,z) rispetto al sistema di riferimento in cui viene immerso.

Nel server di esempio, l'URL di accesso alle API è: "http://martina.maxmagnus.com/ws".

Sono tre le richieste utilizzate per ottenere/modificare informazioni sull'oggetto aumentato:

- **GET /holograms**

Restituisce la struttura dell'oggetto aumentato e le texture a lui associate in formato JSON.

Esempio della struttura dell'oggetto Json:

```
1  { "indicesNumber": numindeces,  
2    "verticesNumber": numvertices,  
3    "texturesNumber": numtextures,  
4    "vertex": [elements],  
5    "textureCoordinates": [elements],  
6    "norms": [elements],  
7    "indices": [elements]  
8  }
```

- **GET /led**

Restituisce lo stato dell'oggetto.

Un esempio:

```
1  { "ID": "1",  
2    "state": "on",  
3    "posX": "0",  
4    "posY": "0",  
5    "posZ": "0",  
6    "scale": "0.003",  
7    "rotationSpeed": "2"  
8  }
```

- **PUT /led**

Consente di modificare alcuni parametri dell'oggetto aumentato.

“state” indica se il led è acceso o spento,

“rotationSpeed” controlla la velocità di rotazione dell'oggetto aumentato (questa proprietà non è specifica dell'applicazione che stiamo trattando, ma è stata inserita per dimostrare che un'entità aumentata non è limitata esclusivamente a caratteristiche derivate dall'oggetto associato nel mondo reale, ma può aggiungergli del valore informativo).

Un esempio:

```
1 { "state": "on",  
2   "rotationSpeed": "2"  
3 }
```

### 5.3.2 Applicazione client

L'applicazione è stata realizzata in Android Studio e sfrutta la libreria Vuforia per il rendering delle entità aumentate. L'applicazione si preoccupa di gestire la visualizzazione dell'oggetto sul target e fornire alcune modalità per interagire con esso e modificarne lo stato.

Nel caso specifico il target che permette di tracciare l'oggetto, è un'immagine stampata (Figura 5.1).

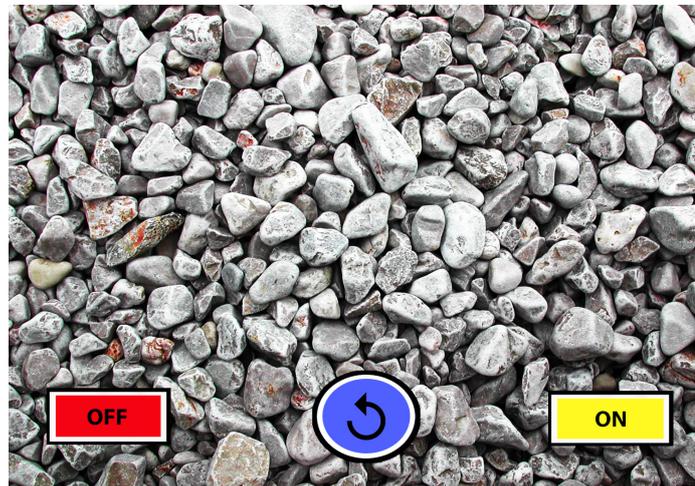


Figura 5.1: ImageTarget dell'applicazione.

L'immagine di sfondo deve essere abbastanza particolareggiata per consentire il riconoscimento da parte della classe Vuforia "TrakerManager" per posizionare l'oggetto aumentato nella corretta posizione e orientamento spaziale rispetto alla telecamera dell'utilizzatore.

Per rendere più semplice l'interazione dell'utente con l'applicazione è stato deciso di disegnare sul marker le corrispondenze ai bottoni virtuali. Questi permettono di spegnere (off) e accendere (on) la luce del led.

Il bottone centrale serve per attivare/deattivare la rotazione del led virtuale.

### Architettura

Le classi principali del software sono mostrate in Figura 5.2:

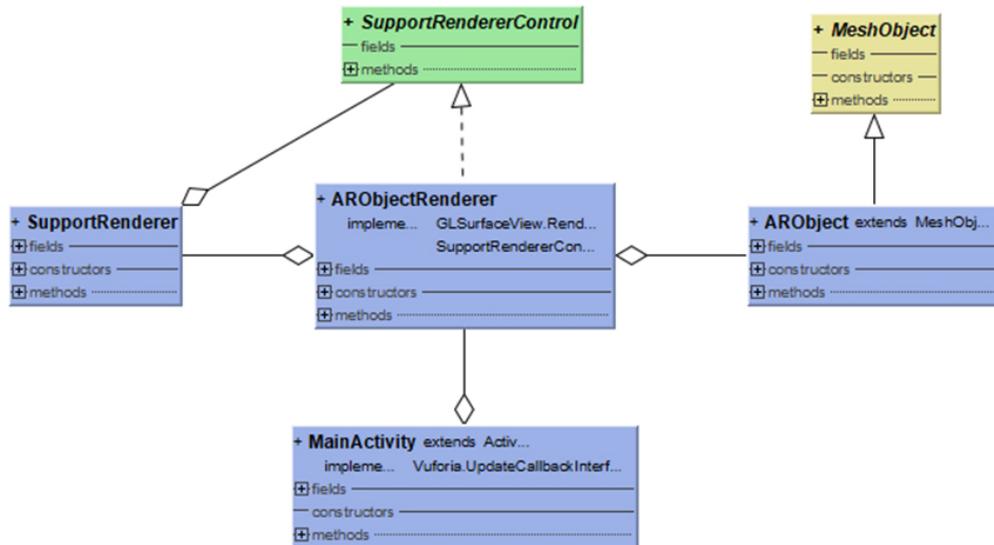


Figura 5.2: Classi principali Android.

La *MainActivity* si occupa di inizializzare Vuforia, caricare l'*object tracker* (il marcatore), inizializzare il renderer e far partire la telecamera del dispositivo. La classe *ARObjctRenderrer* carica l'oggetto dal server (struttura, texture, caratteristiche, ...) e si predispone a visualizzarlo nella corretta posizione spaziale appena riconosce il marker. Si occupa anche dell'inizializzazione e dell'interazione dell'utente con i bottoni virtuali. Inoltre istanzia un timer che ad intervalli

di tempo regolari interroga il server per sapere eventuali modifiche delle caratteristiche dell'oggetto.

E' stata scelta una soluzione a polling perché per effettuare una comunicazione push da parte del server si sarebbero dovuti memorizzare tutti i client connessi e prevedere l'invio di un messaggio per ognuno di essi ad ogni cambiamento dell'oggetto. Questo avrebbe complicato eccessivamente l'implementazione che per questa sperimentazione non era necessario.

La classe *ARObject* rappresenta un oggetto di realtà aumentata generico e non appena vengono reperite le informazioni necessarie alla sua visualizzazione, questo viene creato a *run-time*.

Quando vengono premuti i bottoni on/off, se lo stato del led cambia, il cambiamento di illuminazione è reso con lo switch tra due texture e una chiamata PUT al server imposta lo "state" dell'oggetto ad "on" oppure ad "off". Il terzo bottone è il "rotate" e serve per avviare la rotazione dell'oggetto aumentato.

Durante lo sviluppo del progetto è stato utilizzato il debugger di Adroid Studio.

## 5.4 Ipotesi di un framework per WoAT

Questa applicazione di realtà aumentata nasce dall'esigenza di sperimentare l'idea di WoAT interfacciandosi con un middleware per WoT.

Le funzionalità implementate avevano lo scopo di sperimentare i vari aspetti della rappresentazione di un mondo virtuale associato al mondo reale, cercando un primo approccio ai mondi aumentati e all'interazione con essi.

L'applicazione è stata realizzata *ad hoc* senza l'utilizzo di un'infrastruttura per *augmented world* e ciò è stato possibile perché il problema e le necessità progettuali erano relativamente semplici. Nel caso in cui il progetto richieda un maggior numero di oggetti aumentati con più proprietà modificabili, interazioni complesse e la possibilità di utilizzare coordinate geografiche o target multipli, sarebbe auspicabile avere accesso ad un framework che ne agevoli la progettazione, lo sviluppo e la manutenzione.

Questo studio ha evidenziato quelle che sono le peculiarità che dovrebbe avere un framework di supporto ad un sistema WoAT. Nella sezione seguente ne ipotizziamo un esempio che segue il modello descritto nell'articolo [2] esposto al capitolo 4.

Il framework deve consentire la definizione di un mondo aumentato e del suo funzionamento.

Siccome gli oggetti e le loro caratteristiche e funzionalità non sono conosciute a priori, il framework deve essere in grado di gestire delle entità astratte che saranno specializzate dallo sviluppatore al momento della loro implementazione. Allo scopo sarebbe utile disporre di un *descriptor language*, ad esempio JSON, che consenta di descrivere gli oggetti e definire ogni proprietà con relativo tipo, limiti e valori ammessi.

Gli oggetti generati possono essere dei semplici oggetti virtuali o essere accoppiati ad un oggetto fisico.

Il WoAT necessita che le proprietà di più oggetti siano collegate tra di loro. Perciò si definiscono degli agenti specifici che effettuano e controllano questo accoppiamento. Tale soluzione consente anche di gestire il collegamento tra proprietà eterogenee degli oggetti.

Gli agenti software permettono anche di muovere gli oggetti del mondo nel mondo e di cambiare lo stato delle loro caratteristiche/proprietà. Questi agenti reagiscono in modo asincrono ad eventi generati in concomitanza a delle variazioni nella realtà fisica o a dei cambiamenti di stato di entità aumentate, applicando mutamenti nel mondo aumentato a seconda della sua logica di funzionamento.

Quindi per visualizzare un mondo che è in continua evoluzione bisogna che ci siano delle funzionalità che permettano di creare/modificare/trasformare entità aumentate attraverso degli agenti a *run-time*.

Il framework deve consentire la definizione delle entità aumentate e degli agenti che interagiscono con loro.

Lo sviluppatore sfruttando queste due peculiarità definisce il proprio mondo aumentato e tutte le interazioni del mondo con gli oggetti e degli oggetti tra di loro, in questo modo il mondo funziona autonomamente e in modo indipendente dalla tecnologia che si usa per visualizzarlo.

Una volta definito il mondo aumentato, l'applicazione client dovrà interfacciarsi all'infrastruttura attraverso chiamate REST-full Api e visualizzare la parte di mondo che è nel suo campo visivo, ovviamente avendo a disposizione delle tecnologie che gli permettano di sapere la sua posizione rispetto al sistema di riferimento.

La rappresentazione visuale degli oggetti aumentati è completamente svincolata dalla loro esistenza nell'infrastruttura, infatti potrebbe essere gestita da qualsiasi tecnica di visualizzazione, come dispositivi per realtà aumentata (*smart glasses*), pagine web, ecc...

A titolo di esempio, consideriamo un termometro definito come una *thing* nel WoT e la sua corrispondenza ad un'entità aumentata, definita come una *aug-*

*mented thing* nell'ottica WoAT. La *thing* aumentata fornisce un 'estensione' del termometro che mantiene il tracciamento delle temperature nel tempo, per essere così visualizzato in un grafico attraverso un ologramma oppure in una pagina web.

Questo per spiegare che l'*augmentation* del termometro esiste indipendentemente dalla sua visualizzazione.

Quest'infrastruttura presuppone due ruoli che possono essere svolti da due team separati: chi si occupa di creare il mondo e il suo funzionamento e chi si occupa di visualizzarlo.

### 5.4.1 Creazione del mondo aumentato

- Definizione del mondo in quanto dimensioni e posizionamento rispetto ad un sistema di riferimento che può essere geografico o relativo ai target immersi nell'ambiente.
- Definizione delle entità aumentate che popolano il mondo e che possono essere puramente virtuali o collegate ad elementi del WoT (*bounding fisico*). Nel caso in cui un' entità consenta l'interazione con l'utente (UI), questo deve essere specificato nella sua definizione così che il sistema di rendering possa fornire i controlli adeguati a consentire detta interazione.
- Definizione degli agenti che implementano il comportamento delle entità. Questi consentono il funzionamento del mondo aumentato.

### 5.4.2 Creazione dell'applicazione di visualizzazione

A questo punto entra in gioco il visualizzatore che sfrutta le interfacce messe a disposizione dall'infrastruttura per ottenere il set di entità aumentate disponibili (eventualmente filtrate dal campo visivo attuale dell'applicazione client) e rappresentarli con il loro ologramma se disponibile.

Nel caso di oggetti con caratteristiche di interazione UI, l'interfaccia di visualizzazione deve intercettare le azioni dell'utente e comunicarle all'infrastruttura, secondo le specifiche definite nell'entità stessa.

### 5.4.3 Infrastruttura applicata al caso di studio trattato

Ipotizzando di aver avuto a disposizione un'infrastruttura simile a questa, durante la realizzazione del progetto descritto nella sezione precedente, si sarebbero dovuti compiere i seguenti passi.

Per quanto riguarda la definizione del mondo aumentato:

- Assumere il centro del mondo con coordinate 0,0,0 rispetto ad un target e di conseguenza dichiarare tutte le posizioni degli oggetti e il loro orientamento in relazione a questo punto centrale.
- Creare l'entità aumentata *phyLed* collegata ad una *thing* fisica che espone la caratteristica 'stato'.
- Definire un'entità *led* in corrispondenza all'origine del mondo virtuale (quindi coordinate 0,0,0). Questo oggetto ha le proprietà 'ologramma', 'velocitàDiRotazione' e 'colore'. In oltre mette a disposizione un'azione 'rotate' che permette di modificare il suo orientamento usando il valore specificato in 'velocitàDiRotazione'.
- Definire tre entità *bottoneOn*, *bottoneOff* e *bottoneRotate*, in differenti *location*, con le proprietà 'ologramma' e 'premuto'. Quest'ultima è specificata come caratteristica interattiva che consente all'utente di intervenire sull'entità.
- Definire 5 agenti con le seguenti caratteristiche:
  - **Agente1**; l'agente si registra per ricevere i cambiamenti di 'stato' del *phyLed*, e modificare di conseguenza la caratteristica 'colore' dell'entità *led*.
  - **Agente2**; l'agente si registra per ricevere i cambiamenti della proprietà 'premuto' del *bottoneOn* e di conseguenza mettere ad 'on' la caratteristica 'stato' di *phyLed*.
  - **Agente3**; l'agente si registra per ricevere i cambiamenti della proprietà 'premuto' del *bottoneOff* e di conseguenza mettere ad 'off' la caratteristica 'stato' di *phyLed*.

- **Agente4**; l'agente si registra per ricevere i cambiamenti della proprietà 'premutato' del *bottonoRotate* e di conseguenza modifica la caratteristica 'velocitàDiRotazione' di *led*.
- **Agente5**; l'agente ad intervalli regolari chiama l'azione 'rotate' di *led*.

Per quanto riguarda l'implementazione dell'applicazione client:

A questo punto il mondo è già funzionante, il visualizzatore deve occuparsi di interrogare l'infrastruttura con chiamate REST-full Api per conoscere le entità presenti nel mondo da visualizzare e gestire le interazioni dell'utente con esse. Saranno gli agenti software ad occuparsi di modificare adeguatamente il mondo aumentato.

## 5.5 Valutazione del risultato

Questa sperimentazione è nata dall'esigenza di valutare le potenzialità offerte dall'utilizzo dei protocolli del WoT negli *augmented worlds*. Inizialmente è stato realizzato il server REST contenente le API delle entità aumentate (*augmented things*) utili a questo esperimento. In questo caso non è stato implementato il framework proposto nella sezione precedente perché avrebbe richiesto più tempo di quello a disposizione, perciò il server funge semplicemente da deposito di dati per la rappresentazione delle entità aumentate e offre le rispettive API per gestirle. Le API per le *things* fisiche invece sono state definite utilizzando un middleware specifico per WoT.

Durante la prima fase di sviluppo dell'interfaccia grafica sono stati utilizzati Vuforia e Unity per la visualizzazione degli oggetti aumentati. Con queste tecnologie pur raggiungendo l'obiettivo ci si è resi conto che l'infrastruttura di Unity appesantiva inutilmente il progetto e poneva dei vincoli implementativi.

Così si è deciso di sfruttare Vuforia su sistema operativo Android, in questa maniera pur dovendosi occupare del rendering degli oggetti (per il quale sono state utilizzate le librerie OpenGL) è risultata più semplice la separazione della logica di funzionamento del modello dalla sua rappresentazione grafica. In questo modo si può agevolmente supportare il pattern model-view-controller.

Siccome certi cambiamenti nello stato delle *augmented things* sono stati implementati nel codice dell'applicazione client, si è avuto conferma che sarebbe stato più corretto nell'ottica WoAT avere degli agenti in esecuzione sul server (che

rappresenta l'*augmented world*) per gestire alcune funzionalità degli oggetti aumentati e delle loro interazioni.

In conclusione si può affermare che l'approccio WoAT è propedeutico allo sviluppo di *augmented worlds* e con l'opportuna infrastruttura unita alle nuove tecnologie di realtà aumentata, è possibile realizzare mondi aumentati aperti sempre più complessi e con una maggiore integrazione tra le cose del mondo fisico e le cose virtuali.



# Conclusioni

In questa tesi si è voluto mostrare un primo esempio di quella che potrebbe essere un'applicazione di WoAT in cui si integrano diversi ambiti tecnologici. Per questo è stato affrontato in primo luogo un esame delle principali tecnologie utilizzate sul panorama applicativo odierno come Internet delle cose, WoT, dispositivi *wearable* e realtà aumentata.

In merito a questo si è proposto il concetto di Web of Augmented Things del quale sono stati descritti gli elementi essenziali e le potenzialità. Attraverso il progetto realizzato si sono mostrati alcuni aspetti salienti dell'idea di WoAT applicati ad un caso specifico. In seguito è stata esposta un'analisi più generale del problema per fornire un riferimento su come si potrebbe sviluppare questo tipo di sistemi in modo agevole. A tale scopo si è fornita una descrizione dell'architettura di un framework di supporto al WoAT.

Si ritiene che questo nuovo paradigma possa apportare importanti benefici all'ingegnerizzazione di *augmented worlds* e, più in generale, nei sistemi che integrano tecnologie di *pervasive computing* e realtà aumentata. Per convalidare l'idea sono necessarie una vasta gamma di esplorazioni e a tale scopo possono essere prese diverse direzioni valide. L'implementazione di un primo prototipo di piattaforma/infrastruttura come descritta nel capitolo 5 potrebbe essere di sicuro un passo importante per distribuire concretamente i primi AW che integrano WoT.



# Ringraziamenti

Ringrazio il Prof. Alessandro Ricci e l'Ingegnere Angelo Croatti per avermi dato l'opportunità di realizzare questo progetto e per tutto quello che ho potuto apprendere durante il periodo di tesi.

Ringrazio i miei amici, Mattia e Nicola, con i quali ho condiviso i momenti più difficili e anche quelli più belli di questo primo traguardo accademico.

In modo speciale voglio ringraziare la mia fantastica famiglia per aver creduto in me e senza la quale tutto ciò non sarebbe stato possibile. In particolare i miei genitori, Massimo e Federica, che mi hanno dato coraggio e mi hanno sempre sostenuta mostrandomi il lato positivo di tutte le situazioni.

Ringrazio mio fratello Leonardo, che dalla nostra nascita e anche da prima, è sempre stato al mio fianco e ha affrontato insieme a me ogni gioia, debolezza e pazzia.



# Bibliografia

- [1] Vlad Trifa. Towards the WOT Manifesto, 2009.
- [2] A. Croatti and A. Ricci. Towards the Web of Augmented Things, 2017.
- [3] affariitaliani.it. Il marketing virale non esiste, 2010.
- [4] WhatIs.com. What is embedded system?, 2014.
- [5] M. M. Zanjireh, A. Shahrabi, H. Larijani. "A New Clustering Algorithm for Wireless Sensor Networks", 2013.
- [6] Chai Keong Toh. Ad Hoc Mobile Wireless Networks. United States: Prentice Hall Publishers, 2002.
- [7] iOS 7: informazioni sui servizi di localizzazione - Supporto Apple, 2015.
- [8] G. Dominique and V. Trifa. Building the Web of Things, 2015.
- [9] G. Dominique. Web of Things Application Architecture: Integrating the Real World into the Web, 2011.
- [10] Tein-Yaw Chung, I. Mashal, O. Alsaryrah, V. Huy, W. Kuo, D. P Agrawal. Social web of things: A survey in Parallel and Distributed Systems (ICPADS), International Conference on pages 570–575. IEEE, 2013.
- [11] Vineet Kumar. Customer relationship management. Wiley Online Library, 2010.
- [12] Toni M Somers Arik Ragowsky. Enterprise resource planning. Journal of Management Information Systems, 19(1):11–15, 2002.

- [13] Dr. Stuart D. Cheshire, Dr. Bernard D. Aboba Ph.D., and Erik Guttman. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927, May 2005.
- [14] T. Ylonen and C. Lonvick. The secure shell (ssh) protocol architecture, 2006.
- [15] A. Croatti and A. Ricci. Programming abstractions for augmented worlds, 2015.
- [16] Starner T. Augmented Reality Through Wearable Computing, 1997.
- [17] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays, 1994.
- [18] G. Moreau. Visual immersion issues in virtual reality: a survey, 2013.
- [19] Mourlas and Constantinos. Cognitive and emotional processes in web-based education: Integrating human factors and personalization: Integrating human factors and personalization, 2009.
- [20] Augmented reality: a class of displays on the reality-virtuality continuum. Vol. 2351. 1995, pp. 282–292.
- [21] Henry Fuchs and Jeremy Ackerman. “Displays for Augmented Reality: Historical Remarks and Future Prospects”. In: Mixed Reality. ISMR '99. Boston, Massachusetts, 1999.
- [22] Ivan E. Sutherland. “A Head-mounted Three Dimensional Display”. In: Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I. AFIPS '68 (Fall, part I). San Francisco, California: ACM, 1968, pp. 757–764.
- [23] Thomas A. Furness. The application of head-mounted displays to airborne reconnaissance and weapon delivery. Wright-Patterson Air Force Base, Ohio, USA, 1969.
- [24] Thomas A. Furness. “The Super Cockpit and its Human Factors Challenges”. In: Proceedings of the Human Factors Society Annual Meeting 30.1 (1986), pp. 48–52.

## *BIBLIOGRAFIA*

---

- [25] Henry E. Lowood for The Encyclopedia Britannica. Virtual reality (VR).  
URL: <https://www.britannica.com/technology/virtual-reality>
- [26] J. Lanier. A Vintage Virtual Reality Interview. Ed. by Whole Earth Re-view.  
1988. URL: <http://www.jaronlanier.com/vrint.html>.
- [27] Charles B Owen, Fan Xiao, and Paul Middlin. "What is the best fiducial?"  
In: *Augmented Reality Toolkit*, 2002
- [28] Rudolph E. Kalman. "A New Approach to Linear Filtering and Prediction  
Problems", 1960.
- [29] Simon S Haykin. "Kalman filtering and neural networks", 2001.
- [30] Hugh Durrant-Whyte and Tim Bailey. "Simultaneous localization and  
mapping: part I", 2006.
- [31] —, "Project Glass: An Extension of the Self," *Pervasive Computing*,  
IEEE, vol. 12, no. 2, pp. 14–16, April 2013.
- [32] A. Omicini, A. Ricci, and M. Viroli, "Artifacts in the A&A meta-model for  
multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol.  
17 (3), 2008.