

In che modo lo IoT (Internet of Things) influenzerà i tester?

In una serie di articoli Paul Gerrard, guru del testing e consulente, discute vari argomenti collegati all'ambito del testing. In questo articolo, Paul illustra come l'Internet of Things (IoT) cambierà completamente le cose per i tester e il testing. Paul introduce un'architettura che si compone di sette livelli e discute la portata dello IoT e la gamma di problemi che crea per il testing.

Introduzione

In questa serie di articoli, vorrei analizzare l'impatto che lo IoT, anche detto Internet of Everything (IoE), avrà sul testing e sui tester.

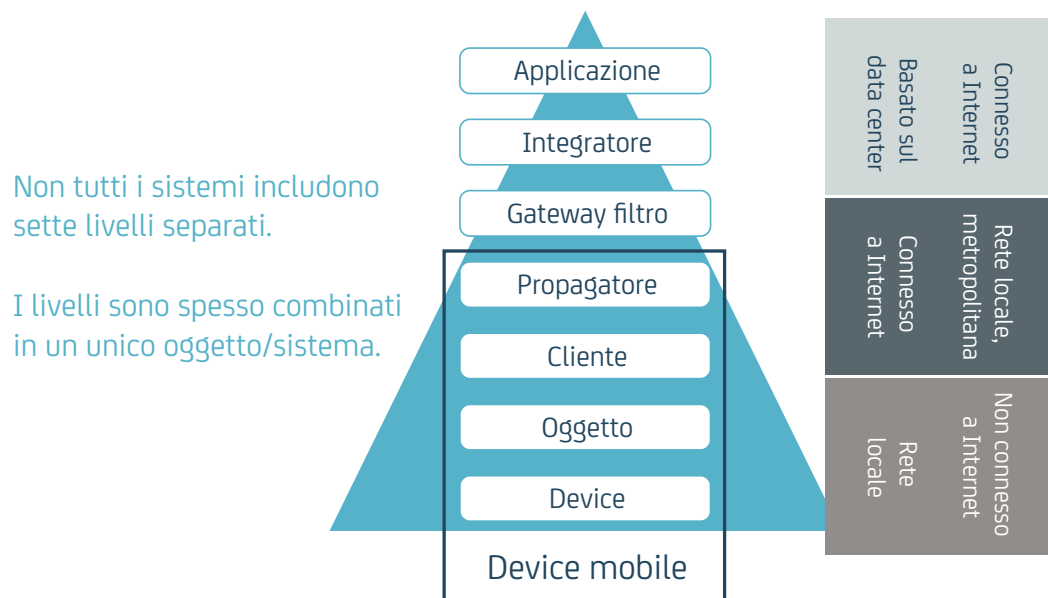
Sembrerebbe che tutto, prima o poi, sia destinato a diventare "intelligente". In parallelo con le applicazioni domestiche, settori come la vendita al dettaglio, la produzione, i trasporti, l'agricoltura e le telecomunicazioni stanno abbracciando lo IoT con entusiasmo. Inutile dire che, il governo e l'esercito fanno la loro parte con la ricerca; e, naturalmente, fenomeni emergenti, le smart city, veglieranno o controlleranno la nostra vita quotidiana.

Questo articolo fa riferimento a una serie di pezzi che ho scritto [1, 2, 3] e trae alcune conclusioni circa l'impatto su testing e tester.

Un modello stratificato di IoE

Nello stesso modo in cui gli standard per i device di rete alla periferia stanno evolvendo, non esiste uno standard o un'architettura convenzionale per lo IoE. Tuttavia, stanno emergendo alcuni schemi e, come ci si potrebbe aspettare, l'architettura può essere suddivisa in livelli logici.

Architettura a 7 livelli



Questo modello a sette livelli è un amalgama di diversi stili di creazione di schemi pubblicati; è neutrale rispetto a device e prodotti e non è allineato con una particolare tecnologia. In termini di scala e di prospettiva, questi livelli sono meglio rappresentati come gerarchia, con il livello dei device alla base. I sette livelli sono descritti in [2]. Indipendentemente dall'implementazione, il modello a livelli dovrebbe aiutare a capire l'architettura del sistema.

I rischi dell'insuccesso

Questa architettura a sette livelli può aiutare a capire la funzione di ciascun componente in un'implementazione IoE. Ho utilizzato il modello a livelli per creare un elenco di quelli che chiamerei modelli di rischio [2].

Ma esistono anche rischi sociali o personali che vengono messi in evidenza dai media, e ai quali dobbiamo prestare una certa attenzione. Questi possono esistere o meno per la tua applicazione; ma, se esistono, saranno probabilmente specifici per il progetto. Ecco allora i candidati principali:

- **Rischi sociali/personali:** a dominare sono sicurezza e privacy.
- **Complessità:** le interazioni tra i device possono essere imprevedibili, impreviste e sconosciute.
- **Privacy:** la raccolta dei dati è pervasiva ma invisibile, e in gran parte fuori dal nostro controllo.
- **Abusi:** lo IoT porta benefici, ma la pervasività delle reti e dei dati agevola attività criminali.
- **Sicurezza aziendale:** sistemi aziendali una volta sicuri sono ora collegati a reti molto più grandi e meno protette.

In contesti quali sistemi di trasporto, linee di produzione manifatturiere, stazioni televisive, generazione/distribuzione di energia e città intelligenti, il potenziale per hacking, interruzioni e azioni terroristiche è illimitato. Il settore della sicurezza ha molto da imparare e ancora più da fare. Lo IoT è un ambito completamente nuovo.

Lo IoE apporta nuovi livelli di complessità e di scala. I rischi non funzionali sono ragionevolmente noti e sappiamo come affrontarli. La novità è la necessità di eseguire test funzionali e simulazioni su larga scala.

La portata dei testing IoE

La gamma delle problematiche generate dallo IoE è più ampia che mai. Non tutti i sistemi IoT saranno enormi, complessi e costosi: ma tutti forniranno un profilo tecnico e di rischio diverso da quelli cui siamo abituati. Ecco allora le dimensioni principali dell'ambito per il tester IoT:

- **Test a livello di hardware:** i device al livello più basso sono sofisticati, ma svolgono funzioni semplici; la maggior parte delle quali verrà eseguita dai produttori.
- **Scala:** un'applicazione indossabile può essere semplice nella sua architettura, ma verrà installata potenzialmente da una base di utenti di milioni di persone.
- **Funzionalità a livello di oggetto e di server:** l'attività di testing funzionale si svolgerà per la maggior parte a livello di hub locali, aggregatori e server basati su data center. Le architetture andranno da semplici web app a sistemi con decine di sottosistemi.
- **Oggetti mobile:** questi entrano ed escono dalla portata delle reti e si spostano tra reti diverse. Le condizioni ambientali, le fonti dei dati e la posizione del device ne influenzano il comportamento. Potenza, interferenze, forza del segnale, roaming e disturbi avranno tutti un impatto.
- **Reti mobili:** alcuni sistemi (ad esempio, le automobili connesse) portano con sé una propria rete locale. Una rete mobile incontrerà altre reti che creano interferenze o, potenzialmente, reti non autorizzate o non sicure, ponendo problemi di sicurezza.
- **I rischi per la sicurezza di rete esistono a vari livelli:** device non autorizzati utilizzano la tua rete e intercettano i dati, o ne inviano di falsi. Access point non autorizzati assumono il controllo delle connessioni e dei dati degli utenti. Le vulnerabilità esistono a tutti i livelli all'interno della tua architettura, e sono soggette ad attacchi.

- **Registrazione del device, provisioning, malfunzionamenti e sicurezza:** registrazione iniziale del device e provisioning sono suscettibili di malfunzionamenti. I device sono soggetti a interruzioni di alimentazione e all'azione di neve, caldo, freddo, vandali, animali, ladri e così via. È possibile che debbano essere testati processi automatizzati di spegnimento, accensione e autenticazione, configurazione e registrazione.
- **Confusione collaborativa:** device in movimento (di nuovo, le automobili) collaborano in modi complessi e in numeri elevati. Ma gli incidenti capitano, i conducenti cambiano idea, i parcheggi si liberano e vengono occupati in modo casuale, e quindi l'algoritmo di ottimizzazione deve gestire situazioni in rapida evoluzione. Allo stesso tempo, questi servizi non devono confondere gli utenti.
- **Integrazione a tutti i livelli:** flussi semplici e complessi di dati e controllo, end-to-end.
- **Big Data: logistica:** del sistema da testare faranno parte massicci servizi di storage di dati.
- **Big Data: analisi e visualizzazione:** la scienza e la visualizzazione dei dati sono suscettibili di inclusione nel campo di applicazione dello IoT. Questo include dati tempestivi, accurati e coerenti, nonché filtraggio, unione, integrazione e riconciliazioni.
- **Privacy personale e aziendale:** hacker e truffatori sono una minaccia, ma anche i vari governi possono essere visti come potenziali malintenzionati.
- **Device wearable e embedded:** device wearable e human-embedded presentano sfide nuove e specifiche.
- **Connessione totale:** verrà un momento in cui tutti i device utilizzati, ad esempio, in un ospedale, albergo o stabilimento, saranno connessi.

La gamma delle problematiche che dobbiamo prendere in considerazione per il testing dello IoE si è ampliata, così come la scala dei test richiesti.

Test funzionali su larga scala

Quando testiamo la funzionalità di componenti a un livello superiore dell'architettura, soprattutto a livello di integratori e di applicazioni, potremmo dover simulare migliaia o milioni di device sul campo. Il numero di combinazioni e permutazioni potrebbe essere superiore a qualsiasi capacità di calcolo o previsione. Le nostre simulazioni andranno a generare ripetutamente scenari da testare, a registrare i risultati e potenzialmente a ripetere le simulazioni per studiarle in un secondo momento.

I componenti di livello superiore devono essere testabili. Avremo bisogno di elementi come gestori di eccezioni, utilità che inseriscono dati, acquisiscono e replicano o riproducono scenari. Cem Kaner ha scritto parecchio di quello che definisce "high volume automated testing" (testing automatizzato a volumi elevati) [4]. Questo è un buon punto di partenza.

Queste tecniche potrebbero anche essere definite testing di Big Data. Dovremo trovare dati adatti al nostro scopo. Dovremo generare, contrassegnare, modificare ed eseguire il seed dei dati, in modo da poterne tracciare l'utilizzo. Avremo bisogno di strumenti per monitorare l'utilizzo dei dati contrassegnati e della capacità di riconciliare i dati derivanti da raccolta, storage, utilizzo e smaltimento. Avremo bisogno di nuovi strumenti di visualizzazione di test per supportare diagnostica e debugging.

Tutto gira intorno ai volumi. I singoli test possono essere importanti o meno, ma la gestione di risultati, visualizzazioni e processi decisionali su larga scala richiederà molto tempo.

Ambienti di test, testing sul campo

In un caso semplice, un laboratorio di test per un prodotto di domotica potrà essere installato in qualsiasi ufficio, dato che la portata della rete locale sarà limitata a una singola famiglia. Nel caso di un sistema di gestione ambientale urbana che controlla i livelli di inquinamento atmosferico di un'intera città, ad esempio, l'acquisizione dei dati del sensore potrebbe essere simulata in laboratorio. Tuttavia è probabile che si debba eseguire un pilota del servizio in un ambiente urbano reale, per calibrare i sensori, l'aggregazione dei dati e i processi di integrazione, allo scopo di ottenere visualizzazioni di dati significative.

Supporto per gli strumenti

Anche la necessità e l'opportunità di eseguire test manuali esisteranno sempre, una percentuale decisamente maggiore di attività di testing, rispetto a quella cui siamo abituati, dovrà essere eseguita mediante strumenti, ai quali sarà affidato un numero molto elevato di test. La sfida non è rappresentata dalla necessità degli strumenti per eseguire i test; ma dalla progettazione delle centinaia, delle migliaia o dei milioni di test da dare in pasto agli strumenti stessi.

Tutti i device attualmente disponibili presentano sfumature e complessità proprie, e sperimenteranno eventi imprevisti. Anche per un sistema semplice gli scenari possibili potrebbero essere nell'ordine delle migliaia. I sistemi sono sempre più complessi, e devono essere testati. Non c'è alternativa.

Analisi di test, visualizzazione e processo decisionale

Illustrerò in maggiore dettaglio le analisi di test in un altro articolo della serie. Se i sistemi di produzione sono piattaforme per la sperimentazione, il codice che serve agli esperti di marketing può anche servire a sviluppatori e tester. Pensa al tuo codice di analisi come a una rete di sensori. Anche i processi DevOps sono "cose".

Test di performance e dati di test

I test di performance, in un mondo IoT, non sono molto diversi da quelli della nostra esperienza attuale. La situazione può complicarsi quando i dati acquisiti dai sensori devono essere coerenti. Ad esempio, i messaggi ricevuti dalle auto all'interno del territorio urbano devono corrispondere a una posizione fisica per essere significativi. Un insieme casuale di coordinate geografiche non servirà a nulla: avremo bisogno di insiemi di dati affidabili derivanti dal mondo reale, o di utility in grado di generare dati significativi e affidabili per i test.

Il futuro per il testing e i tester

Mi pare che livelli elevati di automazione dei test siano destinati a diventare una necessità, perché lo IoE si trasformi in realtà. L'automazione servirà non a facilitare i test, ma a renderli possibili.

L'automazione dei test in un ambiente DevOps è una fonte di dati per l'analisi proprio come i sistemi di produzione, quindi tecniche e strumenti di analisi costituiscono un altro ambito in crescita.

I tester dovrebbero imparare a scrivere codice? La risposta è semplice: sì. In questo momento è possibile che il tuo lavoro non lo richieda, ma gli annunci di lavoro, negli Stati Uniti e in Europa, tendono sempre di più a richiedere capacità di codifica e altre capacità tecniche. Sempre più spesso ti verrà richiesto di scrivere le tue utility, e di scaricare, configurare e adattare strumenti open source, o creare test automatizzati.

Lo IoT e i trend digitali stanno spingendo il testing "a sinistra": apparentemente ogni azienda sta perseguendo quello che viene comunemente definito un approccio "shift-left". Le attività di test o, meglio, le attività di progettazione dei test vengono anticipate all'interno del processo di sviluppo. Il mio consiglio è di abbracciare questa tendenza.

Riepilogo

Lo IoT amplia il testing in termini di scala, diversità e complessità.

Il testing dei componenti di basso livello o dei sottosistemi rimane praticamente invariato rispetto a prima; ma, a livello di sistema, avremo bisogno di metodi di simulazione e automazione dei test a volume elevato. Gli strumenti di cui abbiamo bisogno potrebbero non esistere ancora: per cui, in attesa che i fornitori di strumenti si mettano al passo, potresti trovarti a doverne creare di tuoi.

L'automazione di test a volume elevato richiede modelli di test, generatori di dati di test e Oracle automatici. Modellazione, simulazione, analisi, visualizzazione e processo decisionale mediante strumenti diventeranno competenze importanti per gli architetti di test e i team di test. I tester dovranno imparare a creare modelli di test migliori e a utilizzarli con più strumenti di modellazione e di simulazione tecnica.

La creazione di ambienti di test affidabili e di dati di prova significativi sarà (come sempre) causa di notevoli problemi.

Saranno necessari ambienti di test su larga scala in laboratorio e sul campo, e i confini tra i test in produzione e in laboratorio si confonderanno. Le analisi di test derivate dai processi DevOps diventeranno una disciplina fondamentale per il testing nell'ambito IoT.

L'autore

Paul Gerrard è un consulente, formatore, autore, webmaster, sviluppatore, tester, relatore, allenatore di canottaggio ed editore. Ha svolto incarichi di consulenza su tutti gli aspetti del test e del QA software, specializzandosi nella test assurance. Partecipa con presentazioni e tutorial a conferenze sul testing in Europa, USA, Australia, Sud Africa e, occasionalmente, ha anche vinto dei premi grazie a questi interventi.

Ha studiato presso l'università di Oxford e l'Imperial College di Londra; nel 2010, ha vinto l'Eurostar European Testing excellence Award e, nel 2013, l'European Software Testing Awards (TESTA) Lifetime Achievement Award.

Nel 2002, Paul ha scritto "Risk-Based E-Business Testing" insieme a Neil Thompson. Ha scritto "The Tester's Pocketbook" nel 2009. È stato co-autore di "The Business Story Pocketbook" con Susan Windsor nel 2011 e ha pubblicato "Lean Python" nel 2014.

Sempre nel 2014 Paul è stato Programme Chair della Conferenza EuroSTAR di Dublino.

È Principal di Gerrard Consulting Limited, direttore di TestOpera Limited e host del Test Management Forum.

E-mail: paul@gerrardconsulting.com

Twitter: [@paul_gerrard](https://twitter.com/paul_gerrard)

Web: gerrardconsulting.com

Per ulteriori informazioni, visita **Sviluppo e test** con CA Technologies.



Entra in contatto con CA Technologies all'indirizzo ca.com/it



CA Technologies (NASDAQ: CA) crea software che promuove l'innovazione all'interno delle aziende, consentendo loro di cogliere le opportunità offerte dall'application economy. Il software rappresenta il cuore di qualsiasi business, in ogni settore. Dalla pianificazione allo sviluppo, fino alla gestione e alla sicurezza, CA Technologies collabora con le aziende di tutto il mondo per cambiare il nostro modo di vivere, interagire e comunicare, in ambienti mobile, cloud pubblici e privati, distribuiti e mainframe. Per ulteriori informazioni, visita il sito ca.com/it.

Riferimenti

- 1 The Internet of Everything—What is it and how will it affect you?, Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOEWhatIsIt2.pdf>
- 2 Internet of Everything—Architecture and Risks, Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOEArchitectureRisks.pdf>
- 3 Internet of Everything—Test Strategy, Paul Gerrard, <http://gerrardconsulting.com/sites/default/files/IOETestStrategy.pdf>
- 4 An Overview of High Volume Automated Testing, Cem Kaner, <http://kaner.com/?p=278>