

Smart Vineyards with IoT

Michele Contu*,

* Laurea Magistrale in Informatica, Università di Bologna, Italia

Email: michele.contu@studio.unibo.it

Abstract—Il monitoraggio e il miglioramento della produzione agricola giocano un ruolo chiave nel settore dell'agricoltura 4.0, soprattutto per quanto concerne l'agricoltura di precisione, in cui la produzione è ottimizzata al fine dell'ottenimento di un prodotto di qualità eccelsa, tenendo conto altresì delle migliorie apportabili a processi produttivi e decisionali, che incidono direttamente sui costi di mantenimento dell'intero sistema produttivo.

Specialmente nell'ambito della viticoltura, in cui i fenomeni esogeni giocano un ruolo preponderante per la buona riuscita del prodotto, è prioritario munirsi di opportuni sistemi di sensing, che oltre a rappresentare un sistema di monitoraggio diretto, sono utili in una visione lungimirante, all'automatizzazione e raffinazione delle tecniche agricole. In ottica predittiva, i dati ricavati dal sensing possono essere utilizzati per formulare predizioni e predisporre strategie a lungo termine, mitigando così l'effetto delle condizioni ambientali sfavorevoli sulle coltivazioni, e permettendo di agire reattivamente per mitigarle.

Il progetto ha l'obiettivo di mostrare un *proof of concept* di un'applicazione per il monitoraggio di un vigneto, che include un sistema predittivo sulla base dei dati rilevati, quali temperatura, umidità del terreno e bagnatura fogliare. Sono stati introdotti nell'architettura anche due tecnologie emergenti: Web of Things e IOTA.

I. INTRODUZIONE

La relazione andrà a esporre come dalle fasi di ideazione e progettazione si è passati alla realizzazione effettiva dell'applicativo qui presentato. Si è prima di tutto effettuata una fase di ricerca, per attestare quale fosse lo stato dell'arte dell'Internet of Things nell'ambito dell'agricoltura e allevamento di precisione. Si è passati poi alla progettazione del sistema, è stato individuato prima l'hardware con il quale effettuare il sensing, board e sensori, e successivamente i meccanismi e i protocolli di comunicazione con i quali questi comunicano. Conclusa la parte progettuale, ci si è focalizzati sull'implementazione dell'intera piattaforma attraverso le tecnologie software indicate sul capitolo quinto. Infine ci si è dedicati all'analisi dei dati ottenuti dai sensori, che sono stati utilizzati per la formulazione di previsioni, attraverso il metodo Holt-Winters. Il capitolo finale tratta alcune considerazioni e miglioramenti apportabili all'applicativo.

II. PUBBLICAZIONI CORRELATE

Il lavoro qui presente è stato ispirato da una serie di letture che hanno come base il paradigma IoT per il monitoraggio in contesti agricoli e zootecnici. Per ogni paper analizzato sono stati considerati gli aspetti salienti, e riconosciuti i punti di forza di ognuno, ci si è serviti di quanto appreso per confezionare un'architettura personalizzata.

A. Applicazioni IoT nell'agricoltura e nell'allevamento

La pubblicazione che più ricalca quanto realizzato in questo progetto è in parte contenuto sul paper di A. Medela, B. Cendón, L. González, R. Crespo e I. Nevares del 2013 che si intitola *IoT Multiplatform networking to monitor and control wineries and vineyards* [1]. Nel paper citato è proposta un'architettura innovativa basata sul paradigma IoT, che combina differenti dispositivi wireless muniti di sensori, distribuiti per tutta l'area del vigneto analizzato, che analizzando le condizioni climatiche e ambientali determinano quanto questi incidano sulla produzione, tracciando oltretutto la filiera produttiva delle uve trattate dalle aziende vinicole. In relazione al paper di Z. Pang, Q. Chen, W. Han, L. Zheng del 2015 che si occupa di fornire una tracciatura digitale IoT della catena d'approvvigionamento di prodotti agroalimentari [2], si è scelto di replicare in parte quanto descritto nel progetto qui esposto, (seppur in maniera molto più minimale), scegliendo di salvare alcuni dei dati di monitoraggio sulla Blockchain IOTA, rendendoli così non modificabili, sicuri e non ripudiabili.

Sempre nel campo del sensing in ambito agricolo uno studio di A.L. Diedrichs, G. Tabacchi, G. Grünwaldt, M. Pecchia, G. Mercado, F.G. Antivilo (2014) si sono occupati di studiare gli effetti del gelo su differenti coltivazioni monitorate tramite una WSN (wireless sensor network) [3], mentre M.A. Fourati, W. Chebbi, A. Kamoun (2014) hanno realizzato un sistema intelligente per l'irrigazione di uliveti [4], che attraverso i dati ambientali rilevati permette una pianificazione delle operazioni d'innaffiatura. In particolare si sono occupati di registrare temperatura, umidità del terreno, radiazione solare e possibili piogge, di cui hanno fornito una visualizzazione su dispositivi mobile. L'aspetto sorprendente messo in risalto è la frugalità dei costi per la messa in piedi dell'architettura, che è stata confrontata con le più costose soluzioni di un decennio fa.

Per quanto riguarda la coltivazione indoor, ovvero non in campo aperto, S.L. Li, Y. Han, G. Li, M. Zhang, L. Zhang, Q. Ma hanno proposto uno paper che tratta il monitoraggio ambientale delle serre agricole. [5]. Il monitoraggio dell'intera coltivazione era consultabile da remoto, grazie alla combinazione di WSN, rete internet e rete mobile.

Nel 2015, Lukas, W.A. Tanumihardja, E. Gunawan hanno progettato un sistema di monitoraggio del livello dell'acqua a lungo raggio per gli abbeveratoi dei bovini, utilizzando

un WSN basato sui ricetrasmittenti LoRa, consentendo agli allevatori di osservare la disponibilità di acqua per il bestiame anche quando il fienile era dislocato a 1 o 3 km di distanza. [6]

III. PROGETTAZIONE

Prima di procedere con l'imbastitura dell'architettura dell'applicazione, ci si è interrogati preventivamente su quali fossero i requisiti funzionali e di dominio del problema studiato. Ci si è concentrati prima di tutto sull'identificazione di quali fossero i sensori utili al sensing delle coltivazioni vinicole.

A. Sensoristica e device

La sensoristica, installata direttamente sul campo in cui il vigneto è ubicato, fornisce un monitoraggio delle condizioni ambientali in tempo reale. I sensori e le board selezionate per la progettazione della soluzione proposta sono stati scelti considerando il costo ristretto di ognuno di questi.

- Sensore di temperatura DHT22: il sensore in questione misura temperatura e umidità dell'ambiente circostante. Per fare ciò si serve rispettivamente di un sensore capacitivo e un termistore per misurare l'umidità dell'aria circostante, comunicando le sue misurazioni tramite un segnale digitale. I dati prodotti sono inviati direttamente sull'infrastruttura cloud e sono salvati su un database real-time. Nessuna computazione viene effettuata on-board.
- Sensore di umidità del terreno: il sensore di umidità del suolo è costituito da due piastre conduttive che funzionano come una sonda e fungono da resistore. Quando il sensore viene inserito nell'acqua, la resistenza diminuirà e aumenterà la conduttività tra le due piastre. Un esempio di sensore collegato a una board NodeMCU è riportato in figura 1.

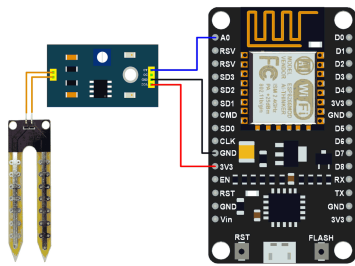


Fig. 1. NodeMCU con sensore di umidità del terreno.

- Sensore bagnatura fogliare: monitora l'umidità della superficie delle foglie, utile per agire proattivamente per l'implementazione di meccanismi automatici di idratazione.
- Sensore RGB: il sensore RGB può essere utile per rilevare tramite il colore dei frutti monitorati il loro grado di maturazione.

IV. ARCHITETTURA

Di seguito sono riportate le scelte stilistiche utilizzate per la realizzazione dell'architettura del progetto. Un esempio completo è riportato in figura 2.

A. Sensoristica

La sensoristica installata on-the-field, come già anticipato, permette di effettuare il monitoraggio di diverse condizioni ambientali e delle colture vitivinicole. Ogni dispositivo, appartenente al proprietario della coltura analizzata, invia i dati su una cloud personale. I sensori sono installati su board NodeMCU munite di Wi-Fi che comunicano con un access point che permette l'accesso a Internet; i dati da questi prodotti sono inviati a un servizio cloud attraverso il protocollo MQTT; la comunicazione avviene in maniera protetta tramite l'utilizzo di certificati X.509. Il sistema di irrigazione è invece gestito tramite una comoda API Web of thing.

B. Cloud Service

Il servizio cloud si occupa di ricevere, smistare e (facoltativamente) salvare i dati ricevuti dalla sensoristica. I dati sono ricevuti tramite il protocollo MQTT, il broker (integrato sulla piattaforma) si occupa di inoltrare i dati ricevuti ai client di ogni proprietario agricolo, in modo che questi siano di facile consultazione. Facoltativamente è possibile per l'utente salvare i dati ricevuti dal client su un registro distribuito simil Blockchain, che garantisce l'infrangibilità e la fidejussione dei dati salvati.

C. Client (proprietari agricoli)

I client dei proprietari agricoli ricevono i dati in tempo reale dai sensori posizionati sul proprio appezzamento. La comunicazione per il reperimento delle informazioni avviene in maniera sicura attraverso un servizio di autenticazione a ruoli integrato sulla piattaforma cloud. I client si occupano a loro volta di inoltrare tramite web socket i dati ricevuti a una API centrale (dislocata anch'essa su cloud). Questa si occuperà di salvare i dati dei sensori (a intervalli regolari) su database real-time e di fornire (congiuntamente al front-end) di strumenti di visualizzazione, analisi e predizione.

D. API centrale

L'API centrale (idealmente istanziata su cloud) si occupa di ricevere con un certo intervallo le informazioni provenienti dai client connessi al broker MQTT. I dati sono salvati su un database real-time pronti per essere analizzati. L'API è protetta anch'essa da autenticazione tramite token JWT. Questa comunica con il front-end e invia dati in tempo reale pronti (tramite web socket) per essere visualizzati (sia dai proprietari delle coltivazioni monitorate che dai clienti di questi).

E. Visualizzazione

L'applicazione front-end permette di monitorare attraverso un'interfaccia utente user-friendly tutte le condizioni ambientali rilevate dai sensori. È possibile visualizzare i dati immagazzinati, creare previsioni e di agire proattivamente per

mitigare le condizioni ambientali sfavorevoli per i vigneti monitorati.

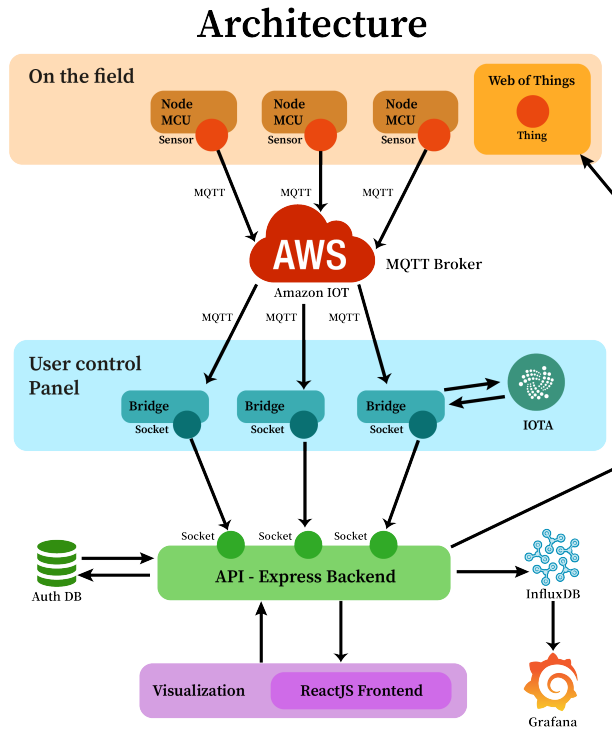


Fig. 2. Architettura generale del sistema

V. IMPLEMENTAZIONE

A. MQTT Communication

La comunicazione tra i sensori, piattaforma cloud e client dei singoli viticoltori è implementata attraverso MQTT. MQTT è un protocollo di messaggistica publish/subscribe utile per l'uso con sensori a bassa potenza trasmissiva.

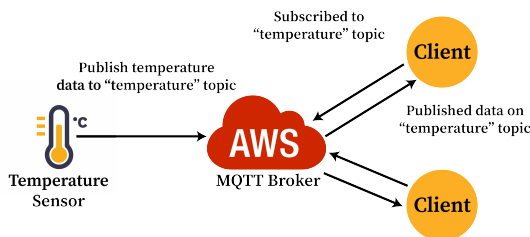


Fig. 3. Esempio di comunicazione MQTT Pub/Sub.

Tre sono i concetti principali da tenere a mente del protocollo MQTT per capirne il funzionamento:

- **Broker:** è una componente che gestisce code per conto di client remoti che si collegano tramite Internet (TCP/IP). I client possono sia pubblicare messaggi testuali nel broker (pubblicatori) e/o possono connettersi al broker ed essere notificati di tutti o alcuni messaggi trasmessi da qualche pubblicatore. In questo caso i client sono sottoscrittori.
- **Topic:** un broker MQTT può essere pensato come il gestore di un numero imprecisato di code, ciascuna

contraddistinta da un nome chiamato appunto topic, cioè argomento. Questo permette ai client sottoscrittori che si conterranno al broker di essere informati solo di alcuni messaggi, appartenenti cioè a quei topic che hanno selezionato al momento della sottoscrizione. Quanto descritto può essere osservato in figura 3. Un client pubblicatore dovrà associare un topic ad ogni messaggio, così il broker capirà se accodare il messaggio in qualche coda, se la coda per quel topic già esiste, o creare una nuova coda per un topic non usato in precedenza.

B. Web of Things

Il Web of Things è uno stile architetturale che permette ai device IoT di essere raggiungibili dal World Wide Web. Nel caso specifico qui riportato, ovvero l'implementazione di un sistema di irrigazione controllabile direttamente da browser, ci si è serviti di una libreria della famiglia Mozilla.

L'API denominata Web Thing fornisce un'interfaccia di programmazione basata su servizi Web con paradigma RESTful per accedere alle proprietà dei dispositivi (things), richiedere l'esecuzione di azioni o per accedere a un elenco di eventi che rappresentano un cambiamento di stato. Le risorse principali che caratterizzano un device nell'architettura Web of Things sono le seguenti:

- **Thing:** la risorsa Thing fornisce una Thing Description per un dispositivo. La Thing è considerata come la risorsa principale (root) del device rappresentato. Esempio in figura 4.
- **Property:** è una risorsa che rappresenta una singola proprietà di un dispositivo. Alcune di queste possono essere solo configurate in sola lettura, mentre alcune possono essere modificabili. Il valore di una Property può essere letto con una richiesta HTTP GET e aggiornato con una richiesta PUT HTTP.
- **Action:** Una Action è una risorsa che rappresenta una coda di azioni relativa a un singolo tipo di azione. Una nuova azione viene creata e inserita nella coda con una richiesta POST HTTP; un elenco di richieste di azioni in coda può essere interrogato con una richiesta HTTP GET.
- **Event:** Una risorsa Event fornisce un registro di eventi recentemente emessi da un dispositivo per un particolare tipo di evento. Solitamente una risorsa Event è impostato in sola lettura.

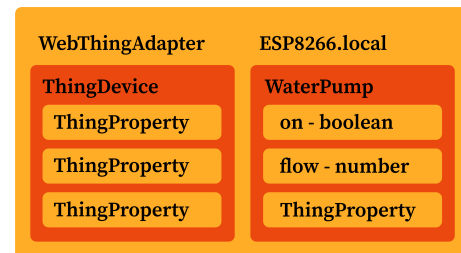


Fig. 4. Una descrizione Web of Thing della pompa per l'irrigazione.

C. Web Sockets

La comunicazione dei dati real-time prodotti dai sensori è trasmessa alla API Express tramite web socket. A sua volta i client scritti in ReactJS ricevono i dati dal backend sempre tramite socket.

Socket.IO è la libreria utilizzata per l'implementazione di quanto illustrato in figura 5, questa abilita comunicazioni bidirezionali in tempo reale e basate su eventi.

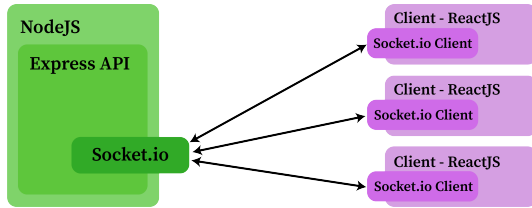


Fig. 5. Comunicazione web socket tra backend e frontend.

La libreria implementa due funzionalità di primaria importanza in un'applicazione come quella presentata:

- Riconnessione automatica: se non diversamente specificato, un client disconnesso tenterà di riconnettersi, fino a quando il server non sarà nuovamente disponibile.
- Rilevamento della disconnessione: tale funzione consente al server e al client di sapere quando uno dei due non partecipa più alla comunicazione.

D. MERN Stack

MERN è un insieme di componenti Open Source che, insieme, forniscono un framework end-to-end per la creazione di applicazioni web dinamiche; partendo dal front-end fino ad arrivare al lato database. Lo stack è composto da:

- MongoDB: database no-sql orientato agli oggetti, utilizzato nell'applicazione descritta per lo storage dei dati di autenticazione per l'utilizzo dell'API Express.
- ExpressJS: è un framework per applicazioni web per Node.js, è stato utilizzato nell'applicazione qui descritta per il back-end.
- ReactJS: è una libreria che abilita la creazione di interfacce interattive che supportano aggiornamenti in real time. Caratterizzato da una struttura a componenti, permette la creazione di interfacce modulari. Ogni componente ha il suo interno uno stato che tiene traccia dei cambiamenti che avvengono a livello di presentazione. Attraverso la modularizzazione delle componenti è possibile (a ogni cambiamento di stato) aggiornare singole componenti, senza dover ricaricare l'intera interfaccia utente.
- NodeJS: è un ambiente di esecuzione per JavaScript, è multiplatforma e orientato agli eventi.

Il backend che si occupa di ricevere i dati dai sensori, ne permette la visualizzazione e il salvataggio sul database realtime è realizzato tramite Express, il frontend tramite React. Il database MongoDB è utilizzato solamente per il salvataggio delle credenziali per l'accesso alle routes del backend.

E. Sicurezza e autenticazione

- Amazon IAM: I sensori inviano dati alla piattaforma AWS (che si occupa principalmente da broker MQTT, non essendo abilitati i servizi di storage supplementari) in maniera sicura tramite l'uso di Amazon IAM. AWS Identity and Access Management consente di gestire in sicurezza l'accesso ai servizi e alle risorse AWS. Grazie ad IAM, è possibile creare e gestire utenti e gruppi AWS e utilizzare autorizzazioni per consentire o negare l'accesso alle risorse AWS.
- Certificati Amazon X.509: A ogni dispositivo (sensori) è assegnato di default da parte di Amazon una coppia di certificati X.509. Tramite l'utilizzo di questi è possibile ricevere i dati da essi inoltrati al broker MQTT. Ogni certificato è univoco e fa riferimento solo a singoli dispositivi registrati tramite Amazon IOT Core.

F. IOTA

Si è scelto di sperimentare la tecnologia IOTA perché molto promettente e molto più scalabile rispetto a una normale Blockchain. Questo risulta ottimale quando si ha la necessità di salvare dati provenienti da sensoristica o per pagamenti machine to machine.

Le caratteristiche di IOTA più pregevoli sono le seguenti:

- Nessun costo di transazione: Per inviare una transazione IOTA, il dispositivo del mittente deve semplicemente verificare due transazioni precedenti nel Tangle, effettuando Proof of Work con difficoltà molto bassa. Nessun costo di transazione, nessuna separazione fra miners e utenti.
- Infinitamente scalabile: dato che per inviare una transazione occorre prima verificarne altre due, all'aumentare degli utenti aumenta anche l'efficienza della rete. IOTA scala proporzionalmente al numero delle transazioni. [7]a
- Transazioni rapide: I tempi di esecuzione delle transazioni sono inversamente proporzionali al numero di transazioni nel Tangle. Quando IOTA raggiungerà un'adozione di massa, le transazioni saranno pressoché istantanee.

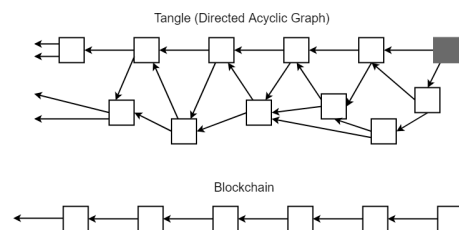


Fig. 6. IOTA Tangle.

G. Real time database e predizioni

Due sono le tecnologie principali utilizzate per lo storing dei dati di sensing e per la visualizzazione grafica di questi ultimi:

- InfluxDB: per lo salvataggio dei dati ci si è serviti di InfluxDB, un database ottimale per il salvataggio di serie temporali. I database utilizzati per lo storing dei dati di temperatura e di umidità hanno una sola measurement (per l'appunto temperatura e umidità) e una sola locazione, ovvero l'appezzamento di terra nella quale i sensori sono disposti.
- Grafana: software di visualizzazione e analisi in coppia con InfluxDB permette di graficare e analizzare i dati prodotti dai sensori, oltre che permettere la formulazione di predizioni.

VI. ANALISI DATI

Come già anticipato, per la visualizzazione dei dati provenienti dai sensori si è utilizzato Grafana. Sempre tramite Grafana e le funzioni di smoothing di Holt-Winters, in esso integrate, si sono potute formulare delle predizioni sui dati raccolti.



Fig. 7. Grafico e predizione dei dati riguardanti l'umidità del terreno.

Come si può vedere in figura 7 è stato predetto l'andamento della serie rappresentante l'umidità del terreno in relazione al tempo.

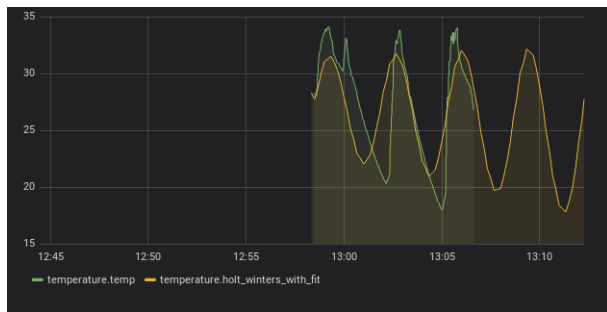


Fig. 8. Grafico e predizione dei dati riguardanti la temperatura ambientale.

In figura 8 si è effettuata una previsione sulla temperatura ambientale. Si vede come, diversamente dal primo caso, in cui la previsione non risultava affidabile, la temperatura in questo caso segua un andamento periodico; la predizione aderisce all'andamento reale e risulta plausibilmente affidabile in relazione alla serie composta dai dati precedenti.

VII. CONCLUSIONI

Il progetto presentato si attiene agli obiettivi che si erano inizialmente prefissati, ovvero la realizzazione di un software di monitoring in ambito vitivinicolo. Tuttavia sono tante le limitazioni del proof of concept prodotto. La prima è sicuramente quella di essere intrinsecamente una piattaforma embrionale, che richiede ancora maggior contestualizzazione in ambito agricolo. La seconda è che la formulazione di previsioni non risulta affidabile, infatti non si possiede né un dataset abbastanza ampio in termini temporali, né si sono utilizzati strumenti predittivi abbastanza potenti. Un possibile miglioramento sarebbe l'implementazione di una parte di progetto dedicata completamente alla predittività attraverso machine learning; a riguardo incluso nello stack TICK, di cui anche InfluxDB fa parte.

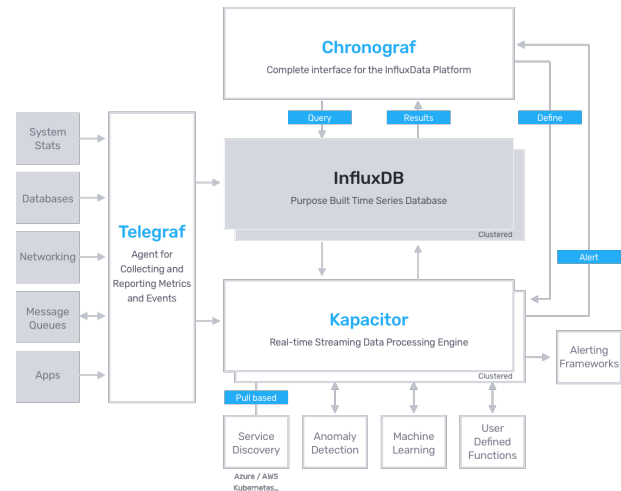


Fig. 9. Esempio di stack TICK.

Una architettura complessiva dello stack TICK è illustrata in figura 9.

REFERENCES

- [1] A. Medela, B. Cendón, L. González, R. Crespo, and I. Nevares, "IoT multiplatform networking to monitor and control wineries and vineyards," in *2013 Future Network Mobile Summit*, pp. 1–10, July 2013.
- [2] Z. Pang, Q. Chen, W. Han, and L. Zheng, "Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion," *Information Systems Frontiers*, vol. 17, pp. 289–319, Apr 2015.
- [3] A. L. Diedrichs, G. Tabacchi, G. Grünwaldt, M. Pecchia, G. Mercado, and F. G. Antivilo, "Low-power wireless sensor network for frost monitoring in agriculture research," in *2014 IEEE Biennial Congress of Argentina (ARGENCON)*, pp. 525–530, June 2014.
- [4] M. A. Fourati, W. Chebbi, and A. Kamoun, "Development of a web-based weather station for irrigation scheduling," in *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pp. 37–42, Oct 2014.
- [5] L. Dan, C. Xin, H. Chongwei, and J. Liangliang, "Intelligent agriculture greenhouse environment monitoring system based on iot technology," in *2015 International Conference on Intelligent Transportation, Big Data and Smart City*, pp. 487–490, Dec 2015.
- [6] Lukas, W. A. Tanumihardja, and E. Gunawan, "On the application of iot: Monitoring of troughs water level using wsn," in *2015 IEEE Conference on Wireless Sensors (ICWiSe)*, pp. 58–62, Aug 2015.
- [7] S. Popov, "The tangle: The official tangle whitepaper," p. 131, 04 2016.