

DIAGNOSTICA DEI SISTEMI: IL CONTROLLO A PORTATA DI CLICK.

Com'è cambiata la capacità di un software di elaborare enormi moli di dati? L'approccio web-based taglia i fili con il passato.

LA CENTRALITÀ DEL DATO.

Quando la richiesta di un cliente è quella di realizzare un prodotto software "chiavi in mano", comincia per noi una grande sfida.

Il principio ispiratore per NIER, che mette il cliente al centro del proprio lavoro, è di accompagnarlo nel processo di definizione di un'idea, che risponda a specifiche necessità, in tutte le fasi, dal Design all'implementazione fino alla validazione.

Per perseguire questo obiettivo, il primo passo è esplorare il parco tecnologico per cercare le soluzioni più adatte, al fine di realizzare il prodotto software richiesto nell'ottica dell'efficienza e dell'affidabilità.

INTRODUZIONE AL PROBLEMA.

NIER è stata coinvolta nella progettazione di un prodotto software in ambito **ferroviario** la cui funzionalità fosse quella di recuperare informazioni diagnostiche e misure elettriche da dispositivi (calcolatori a microprocessore) che regolano la circolazione dei treni (segnalamento ferroviario). L'insieme di queste informazioni deve essere reso disponibile ad un operatore per la consultazione via WEB (attraverso PC e/o tablet). Questo approccio segue la tendenza, manifestata in vari settori anche diversi da quello ferroviario, di abbandonare le applicazioni Stand Alone in virtù di quelle WEB-Based.

Qual è lo scopo di questo software? Aiutare il personale ferroviario nella manutenzione dell'infrastruttura e nell'ottimizzazione dell'efficienza della linea ferroviaria, fornendo in tempo reale lo stato di salute dei dispositivi e permettendo, in caso di guasti, di individuare prontamente il problema e supportarne la risoluzione.

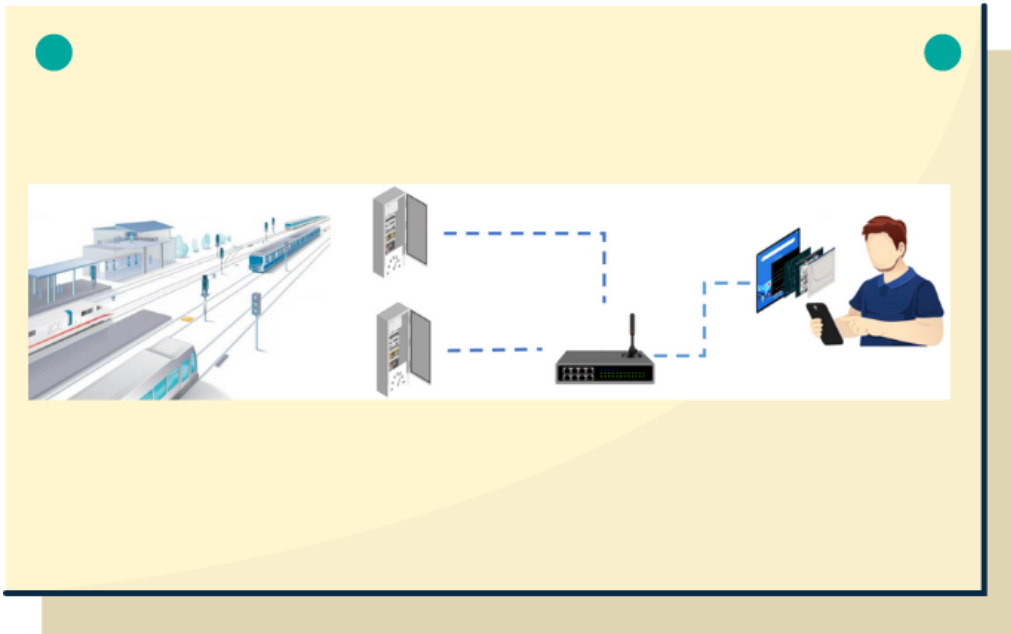
Affinché questo sia possibile, considerando la molteplicità dei dispositivi coinvolti in un impianto di segnalamento ferroviario, il prodotto deve essere facilmente configurabile e distribuibile.

In un sistema di segnalamento ferroviario i dispositivi a Microprocessore comunicano *tramite protocolli di rete* (l'insieme delle regole che definiscono il linguaggio tramite cui essi possono "parlare" e capirsi).

I protocolli di comunicazione possono essere standard oppure **proprietary**, ovvero sviluppati da uno specifico ente o soggetto, come nel caso del nostro cliente. Questo aspetto è una ulteriore complicazione perché richiede uno studio approfondito per conoscere nel dettaglio tutti gli aspetti dello specifico protocollo al fine di poterlo manipolare con padronanza.

Tramite questo insieme di regole, i dispositivi si scambiano ciclicamente informazioni sotto forma di messaggi di rete, con specifici contenuti. Per rendere la comunicazione il più affidabile possibile, in questi sistemi si adotta tipicamente non una sola rete Ethernet bensì due, sulle

quali sono veicolati gli stessi dati, in modo da ottenere la cosiddetta **ridondanza**. Questa architettura consente una maggiore *affidabilità* del sistema (ndr. Certezza di corretto funzionamento).



SOLUZIONE SICURA.

La soluzione proposta è stata lo sviluppo di un modulo Software basato su due diversi livelli:

- Il primo livello, dedicato alla gestione della **ridondanza**, sviluppato in linguaggio C;
- Il secondo livello, dedicato all'implementazione delle regole del **protocollo proprietario** del nostro cliente, sviluppato in linguaggio C++.

I dispositivi di segnalamento forniscono misure di tensione e corrente rilevate sugli enti ferroviari (come segnali, deviatori e circuiti di binario) che sono fondamentali per monitorare lo stato di salute della linea ferroviaria. Oltre a questo, i dispositivi generano automaticamente rapporti/log in caso di arresto anomalo delle apparecchiature (detti DUMP), anch'essi fondamentali per il monitoraggio del corretto funzionamento della linea ferroviaria. Per gli scopi del progetto è necessario gestire entrambi i tipi di informazioni al fine di renderli disponibili per l'elaborazione, sia online che offline.

Una volta recuperate queste informazioni, l'ultimo step è stato quello di visualizzarle in una interfaccia WEB dedicata, accessibile tramite un comune browser Internet (come Google Chrome o Microsoft Edge). A tal fine è stato necessario sviluppare due moduli SW distinti, caratteristici delle applicazioni WEB:

- Modulo di **back-end**, che elabora, classifica ed organizza le informazioni, rendendole disponibili per la visualizzazione;

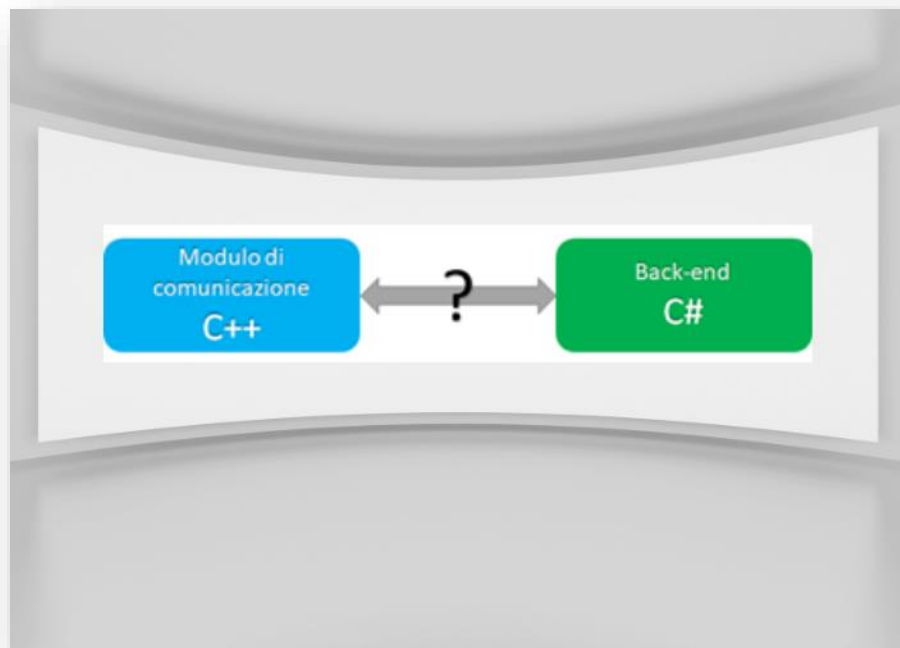
- Modulo di **front-end**, cioè la parte visibile all'operatore, la cosiddetta interfaccia utente con cui quest'ultimo può interagire e visualizzare le informazioni preparate dal back-end.

Entrambi i moduli sono stati sviluppati con il linguaggio C# e la loro comunicazione è stata gestita tramite implementazione di **API REST**, ovvero entità software che permettono la gestione di un meccanismo di interrogazione da parte del modulo front-end, sia in lettura che in scrittura, verso il modulo back-end.

Inoltre, la particolarità dell'applicazione front-end è che doveva rappresentare un'evoluzione di un'applicazione già esistente e fornita dal cliente. Questa, che potrebbe sembrare una semplificazione, in realtà ha comportato un grande lavoro di *reverse-engineering*, per ottimizzare l'integrazione del codice nuovo con quello esistente.

Una caratterizzazione importante di tutto il progetto ha riguardato l'utilizzo di un particolare hardware, fornito dal cliente, basato su sistema operativo open source Linux; con questa premessa, due sono state le domande fondamentali a cui dare risposta:

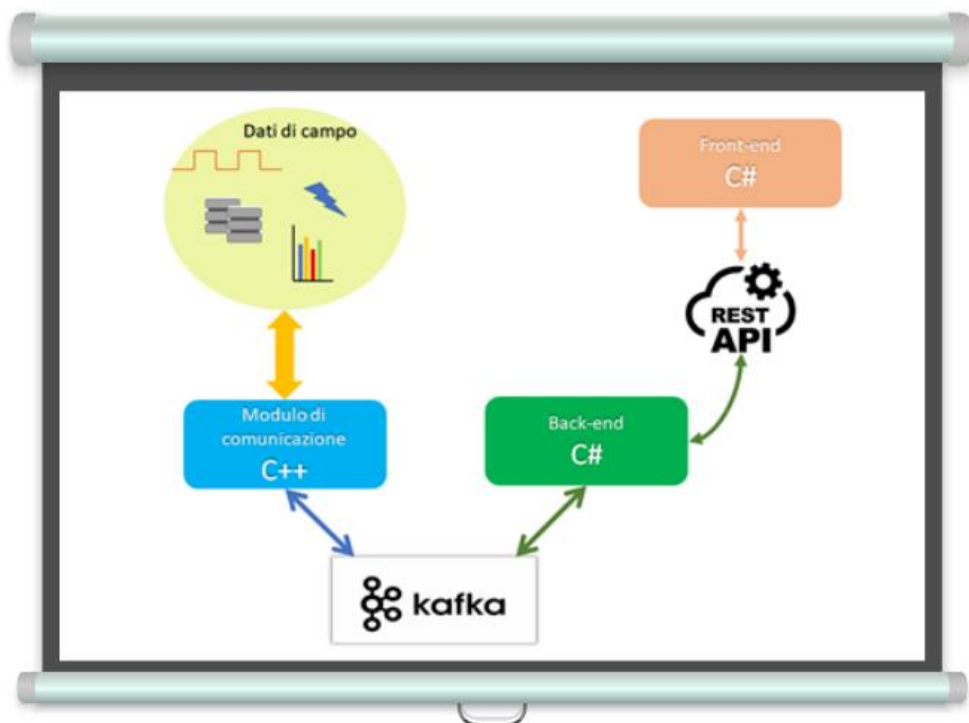
- 1) Quale può essere la tecnologia più efficiente per gestire lo scambio di dati tra il modulo di comunicazione, che implementa il protocollo verso i dispositivi di campo, ed il back-end che deve ricevere ed elaborare le informazioni recuperate?



- 2) quale può essere la via migliore per integrare i tre moduli (comunicazione, front-end e back-end) nell'unico hardware disponibile?



Per quanto riguarda la domanda 1, si è optato per tecnologia Apache **Kafka**, un sistema di messaggistica open-source di pubblicazione e sottoscrizione progettato per fornire una gestione veloce e affidabile dei dati in tempo reale. La grande quantità di informazioni ricevute e la loro alta frequenza sono state correttamente gestite dai due moduli.



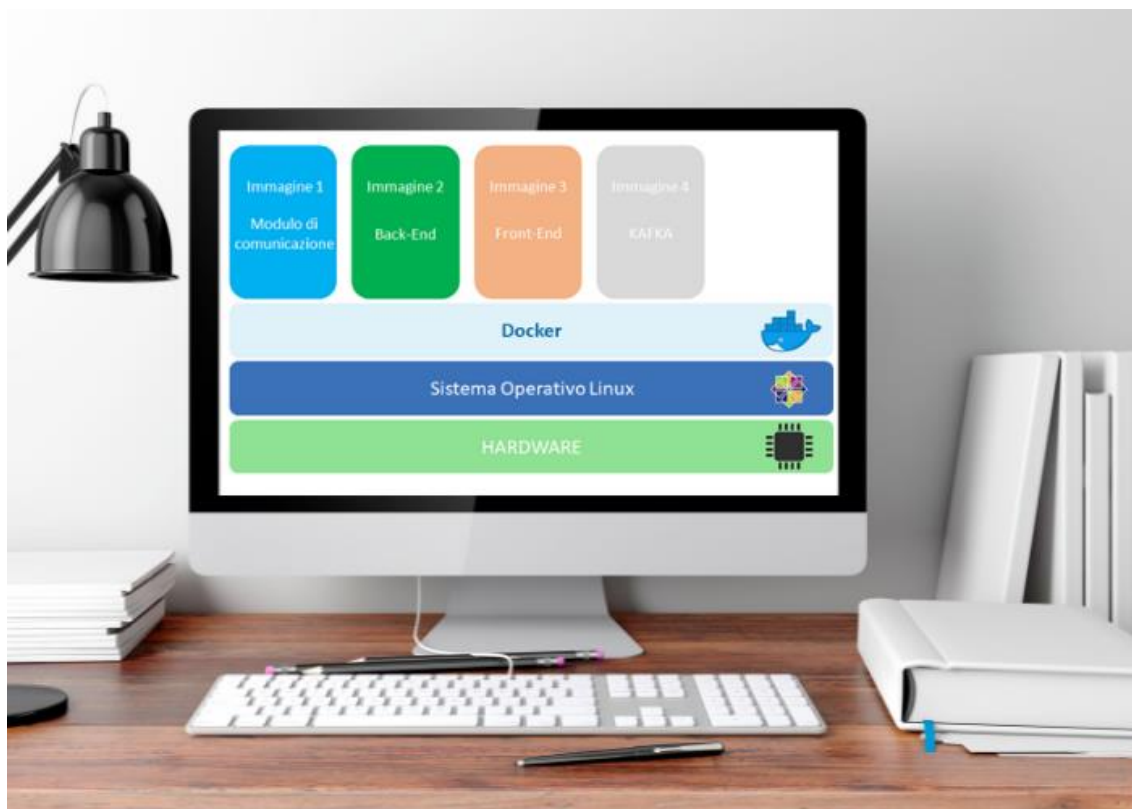
Per quanto riguarda la domanda 2, abbiamo scelto la tecnologia **Docker**, una piattaforma software open-source che permette la creazione di applicazioni auto-consistenti e facilmente distribuibili.

Il concetto alla base di questa tecnologia è la creazione di una “scatola”, detta più propriamente *immagine Docker*, che contiene tutto quello che serve all’applicazione per funzionare e che combina il codice sorgente dell’applicazione con le librerie e le dipendenze del sistema operativo necessarie per eseguire tale codice in *qualsiasi ambiente/sistema operativo*.

Abbiamo, quindi, scelto di creare una *composizione* delle tre applicazioni (Modulo di comunicazione, Back-End e Front-End) in tre distinte immagini Docker, in modo che potessero essere distribuite ed integrate nell’ambiente target (hardware fornito dal cliente) in modo rapido ed automatico.

In aggiunta, avendo anche a disposizione Kafka come immagine Docker, in quanto presente nel registro ufficiale di Docker (<https://hub.docker.com>), è stato possibile includerlo nella fase di integrazione e distribuzione.

Il risultato è stato di ottenere un sistema completamente integrato e funzionante, installando nell’hardware fornito, oltre al sistema operativo, anche la piattaforma software di Docker; in questo modo è stato possibile eseguire le immagini Docker prodotte, integrandole e facendole comunicare automaticamente sia con il mondo esterno (i dispositivi sulla linea ferroviaria da una parte, il PC e il tablet dell’operatore dall’altra), sia tra di loro.



CONCLUSIONE

In questo progetto abbiamo accompagnato il cliente nella definizione del concept del prodotto e lo abbiamo coinvolto nelle evoluzioni della progettazione, dal design e sviluppo, fino alla validazione, in un processo step by step.

Il prodotto software realizzato, caratterizzato da alta specializzazione e massima flessibilità nella sua configurazione, ha permesso al nostro cliente di utilizzarlo con soddisfazione (sua e nostra!) in molti degli impianti da lui installati.