

# **ARCHITETTURA MULTI-TENANT**

---

## **12.1 One Engine, Many Genomes**

### **12.1.1 Codice Sorgente Unico Centralizzato**

L'architettura del Cognitive Twin separa rigorosamente il motore (MMS Kernel) dalla configurazione (Genoma). Esiste un unico codice sorgente del Kernel, manutenuto centralmente, che implementa tutte le capacità di base: ingestione, elaborazione, sicurezza, orchestrazione.

Questo approccio garantisce che ogni miglioramento al Kernel benefici automaticamente tutti i tenant. Una patch di sicurezza viene applicata una volta e protegge tutti. Un'ottimizzazione delle performance accelera tutti i sistemi.

### **12.1.2 Container Segregati per Tenant**

Ogni cliente opera in un container completamente segregato. I dati di un tenant non sono accessibili da altri tenant. Le configurazioni sono indipendenti. I guasti sono isolati.

La segregazione è implementata a livello infrastrutturale, non applicativo. Anche un bug nel codice applicativo non potrebbe causare leak di dati tra tenant, perché i container sono fisicamente separati. File di Configurazione

La personalizzazione avviene interamente attraverso file di configurazione, principalmente il genome.yaml che definisce tutti i livelli del Genoma.

Un nuovo vertical non richiede sviluppo software. Richiede la definizione del Genoma appropriato: quali vincoli di governance, quali competenze di dominio, quali preferenze utente, quali regole contrattuali, quali policy IP.

---

## **12.2 Scalabilità Cross-Industry**

### **12.2.1 Stessa Piattaforma, Verticali Diversi**

La separazione tra Kernel e Genoma permette di servire industrie radicalmente diverse con la stessa piattaforma tecnologica.

Un ospedale utilizza il Cognitive Twin per supportare diagnosi e gestire cartelle cliniche. Una banca lo utilizza per valutare rischi creditizi e compliance normativa. Un'industria manifatturiera lo utilizza per ottimizzare la produzione e la supply chain. Tutti utilizzano lo stesso Kernel; differiscono solo per Genoma.

### **12.2.2 Cambio Configurazione, Non Cambio Codice**

L'espansione in un nuovo settore non richiede mesi di sviluppo. Richiede la definizione del Genoma appropriato, l'ingestione della knowledge base di dominio, la configurazione delle integrazioni necessarie.

Un partner di canale può essere abilitato a configurare nuovi verticali senza accesso al codice sorgente, semplicemente definendo file di configurazione secondo lo schema documentato.

---

## **12.3 L'Azienda "Aumentata"**

### **12.3.1      Organismo Autocorrettivo**

Un'azienda dotata di Cognitive Twin con le appropriate Capsule Gestionali diventa un organismo autocorrettivo: capace di rilevare autonomamente le proprie inefficienze e suggerire correzioni.

Non aspetta l'audit annuale per scoprire i problemi. Non dipende dall'intuizione del management per identificare le opportunità. Il Twin osserva continuamente, confronta con le best practices, segnala le deviazioni.

### **12.3.2      Rilevamento Automatico Inefficienze**

Il rilevamento non è passivo. Il sistema non aspetta che qualcuno faccia la domanda giusta. Cerca attivamente pattern anomali: margini in calo, costi in crescita, ritardi ricorrenti, correlazioni sospette.

Quando identifica un'anomalia, genera un alert proattivo con analisi e suggerimenti. Il management riceve insight actionable, non dati grezzi da interpretare.

### **12.3.3      PMI Competitiva con Multinazionali**

Il risultato finale è la democratizzazione dell'eccellenza operativa. Una PMI con cinquanta dipendenti può operare con processi ottimizzati comparabili a quelli di una multinazionale con migliaia di dipendenti e budget consulenziali milionari.

---

IL divario competitivo basato sulle risorse manageriali si riduce. Conta la qualità delle decisioni, non la dimensione del team che le supporta.