



On-By-One

Valutazione dei capitoli

On-By-One:

Sara Ballardin	2101048
Shuhui Chen	2138001
Simone Giuliano	2145591
Alvise Greggio	2141004
Daniele Nuzzaci	2139039
Federico Temporin	2146764
Tommaso Zorzetto	2148629

Informazioni documento			
Versione	Data	Stato	Destinatari
1.0.0	2026/10/29	Approvato	On-By-One Prof. Tullio Vardanega Prof. Riccardo Cardin

Contatti: onbyone.swe@gmail.com

Registro delle modifiche

Versione	Data	Autore	Descrizione	Verificatore
1.0.0	2026/10/29	Alvise Greggio	Valutazione del capitolato C3	Tommaso Zorzetto
0.1.4	2026/10/29	Tommaso Zorzetto	Valutazione dei capitolati C7 e C9	Simone Giuliano
0.1.3	2026/10/29	Simone Giuliano	Valutazione dei capitolati C2 e C4	Sara Ballardin
0.1.2	2026/10/27	Daniele Nuzzaci	Valutazione dei capitolati C5 e C6	Sara Ballardin
0.1.1	2026/10/26	Sara Ballardin	Valutazione del capitolato C0	Tommaso Zorzetto
0.1.0	2026/10/26	Shuhui Chen	Valutazione dei capitolati C1 e C8	Tommaso Zorzetto

Indice

1	Capitolato scelto: C1 - AURA (Vimar S.P.A.)	3
1.1	Obiettivo	3
1.2	Complessità tecnica	3
1.3	Opportunità di apprendimento	3
1.4	Chiarezza progettuale	4
1.5	Supporto da parte del proponente	4
1.6	Interesse	4
2	Capitolati di interesse	5
2.1	C8 - Sonar (Halue S.r.l.)	5
2.1.1	Obiettivo e tecnologie	5
2.1.2	Considerazioni e criticità	5
2.1.3	Motivazioni dello scarto	5
2.2	C5 - RareCare Navigator (Miriade S.r.l.)	5
2.2.1	Obiettivo e tecnologie	5
2.2.2	Considerazioni e criticità	6
2.2.3	Motivazioni dello scarto	6
3	Altri capitolati valutati	6
3.1	C0 - AIRA (Sanmarco Informatica)	6
3.1.1	Obiettivo	6
3.1.2	Motivazioni dello scarto	7
3.2	C2 - NEXUM Concierge (Eggon)	7
3.2.1	Obiettivo	7
3.2.2	Motivazioni dello scarto	7
3.3	C3 - Fosidot (Uranion EMEA Ltd)	7
3.3.1	Obiettivo	7
3.3.2	Motivazioni dello scarto	8
3.4	C4 - Piattaforma per la gestione dei veicoli e dei servizi post-vendita (M31 S.r.l.)	8
3.4.1	Obiettivo	8
3.4.2	Motivazioni dello scarto	8
3.5	C6 - Sistema AI di caching semantico e dashboard FinOps (Var Group)	8
3.5.1	Obiettivo	8
3.5.2	Motivazioni dello scarto	9
3.6	C7 - SmartDashboard (Ergon Informatica S.r.l.)	9
3.6.1	Obiettivo	9
3.6.2	Motivazioni dello scarto	9
3.7	C9 - SpecCheck (Zucchetti S.P.A.)	10
3.7.1	Obiettivo	10
3.7.2	Motivazioni dello scarto	10

1 Capitolato scelto: C1 - AURA (Vimar S.P.A.)

1.1 Obiettivo

Il progetto consiste nella realizzazione di *AURA* (AI for Understanding, Reasoning and Automation), un'interfaccia che permette di controllare e interrogare in linguaggio naturale un impianto domotico Vimar View Wireless, una rete *mesh* Bluetooth, tramite servizi di intelligenza artificiale commerciali come Claude, ChatGPT, Copilot, Gemini. La soluzione si fonda su un *MCP Server* e rispettivo client, che fanno da interfaccia tra il servizio AI e l'API KNX IoT 3rd-party, esponendo in modo standardizzato dati, funzionalità e comandi dell'impianto.

1.2 Complessità tecnica

Il progetto presenta un livello di complessità medio-alta. Lo sviluppo riguarda prevalentemente il back-end, poiché l'utente interagirà con il sistema direttamente tramite l'interfaccia nativa del servizio AI scelto. I requisiti obbligatori si basano su:

- **Infrastruttura cloud-ready:** basata su container Docker e *Infrastructure as Code* (IaC), avviabile con un solo comando (`docker compose up`) e compatibile per un rilascio nel cloud AWS.
- **MCP Server e MCP Client:** progettati per supportare le API KNX IoT 3rd-party (v2.1.0), in particolare Authorization, Datapoint, Function, Information, Location, Subscription, Timeseries.
- **Autenticazione e autorizzazione:** gestiti tramite OAuth2, con supporto per la selezione dell'impianto e operazioni di lettura e scrittura sui dispositivi.
- **Integrazione AI:** collegamento con almeno un servizio AI commerciale e gestione dei casi d'uso base e avanzati indicati nel capitolato.
- **Testing:** raggiungimento di una copertura minima del 75% per i test *unit/integration* e dell'80% per i test *E2E* (opzionalmente sopra il 90%).

1.3 Opportunità di apprendimento

Si interagisce con numerose tecnologie:

- **MCP:** protocollo Model Context Protocol server, con i relativi *tools* e *resources*, e client collegato al server.
- **GenAI:** progettazione di strumenti utilizzabili da un LLM, gestione di richieste ambigue e sinonimi, conferme sulle scritture.
- **IoT:** integrazione con un sistema domotico fisico (Vimar View Wireless) e dispositivi reali (gateway BLE, attuatori, sensori di presenza UWB, termostato), con comprensione della relazione tra gli elementi dell'impianto (Location-Function-Datapoint) di KNX IoT.
- **API e sicurezza:** standard internazionale KNX IoT, OAuth2, gestione multiutente e privacy dei dati verso l'LLM.
- **Cloud e DevOps:** Docker e `docker-compose`, IaC, *Continuous Integration* (GitHub Runners), opzionalmente anche AWS (LightSail, EC2) e Terraform/CDK/Ansible.
- **Backend:** Python con FastMCP (consigliato), in alternativa TypeScript o Java.
- **Database** (se necessario): relazionali (MySQL, PostgreSQL) o Redis per le sessioni.

1.4 Chiarezza progettuale

La documentazione offre una base solida per l'analisi dei requisiti; le esigenze sono suddivise in obbligatorie, opzionali e *nice to have*, i prodotti attesi sono descritti con precisione e gli schemi (macro-componenti, architettura MCP, foto del kit) ne facilitano la comprensione. Le due tabelle di casi d'uso e le risposte date durante la riunione fatta con la proponente hanno chiarito dubbi quali il ruolo del client MCP, la gestione dei token, i *mock* e il *PoC* minimo.

1.5 Supporto da parte del proponente

Il supporto offerto da Vimar S.P.A. è stato uno dei fattori determinanti nella scelta di questo capitolato, per la sua ampiezza e per la chiarezza con cui è descritto. Vimar mette a disposizione:

- *SAL (Stato Avanzamento Lavori)* bisettimanali di un'ora, che dopo il Proof of Concept possono diventare settimanali di mezz'ora, svolti su Microsoft Teams oppure in presenza nelle salette della sede.
- La possibilità di richiedere, con 1–2 settimane di anticipo, approfondimenti mirati (anche con colleghi di altri team) e incontri di allineamento extra su dubbi progettuali, scelte architetturali e coerenza della soluzione.
- Due incontri in presenza obbligatori a Padova, all'inizio (con consegna del kit) e al collaudo finale, oltre a demo con i colleghi R&D a metà progetto (PoC) e a fine progetto (*MVP*).
- Contatti email diretti con i referenti aziendali, con risposte negli orari d'ufficio, per le comunicazioni asincrone.
- La firma digitale dei verbali esterni redatti dal gruppo.
- Il materiale di lavoro: un kit hardware portatile per l'intera durata del progetto, un account amministratore con possibilità di aggiungere utenti, l'accesso alle API e alla documentazione KNX IoT, ed eventualmente, in accordo con la proponente, spazio AWS e una chiave LLM per le prove.

1.6 Interesse

Il gruppo ha mostrato un interesse particolare per questo capitolato, soprattutto per la possibilità di lavorare su un prodotto fisico: poter provare l'impianto reale e vedere concretamente il risultato del proprio lavoro è stato percepito come un elemento distintivo rispetto agli altri capitolati. Tra i temi di maggiore interesse figurano l'integrazione di un ambiente domotico con un MCP Server e un MCP Client, la gestione delle ambiguità nelle richieste rivolte alla GenAI, l'utilizzo di API per il collegamento tra le parti e l'implementazione del protocollo OAuth2 per l'autenticazione e la gestione dei dati sensibili dell'utente. Hanno inoltre suscitato interesse le tecnologie di cloud e DevOps come AWS, **docker compose** ed eventualmente Terraform, CDK e Ansible, molto richieste nel settore dell'ingegneria del software. Hanno pesato in modo rilevante anche la presentazione in aula, l'incontro di approfondimento aggiuntivo, risultati chiari ed esauritivi, l'entusiasmo con cui l'azienda ha coinvolto il gruppo nel progetto e la grande disponibilità dimostrata, anche grazie alla retrospettiva sulle edizioni passate svolta per comprendere come offrire un supporto più efficace.

2 Capitolati di interesse

2.1 C8 - Sonar (Halue S.r.l.)

2.1.1 Obiettivo e tecnologie

Sonar è un'applicazione web di sola lettura creata per collegarsi a uno store Shopify *enterprise*. Il suo scopo principale è analizzarne le configurazioni e il codice tramite la Admin GraphQL API, ricostruendo le dipendenze esistenti e individuando ciò che non è più necessario. L'obiettivo è avere una documentazione sempre aggiornata (consultabile ed esportabile in formati come Markdown e HTML) e un *rapporto di igiene tecnica* focalizzato su *codice morto* e *configurazioni orfane*. Risulta particolarmente utile per i passaggi di consegne tra fornitori e per le fasi di *replatforming*.

Il livello di complessità tecnica del progetto è medio-alto e richiede il rispetto di requisiti rigorosi:

- **Tecnologie vincolanti:** Next.js (App Router) ospitato su Vercel, orchestrazione tramite Vercel Workflows e Cron, base dati Supabase (PostgreSQL) con *Row Level Security*, codice in TypeScript strict e autenticazione esclusiva tramite Google.
- **Funzionalità chiave:** *scansione durevole* e resiliente a *rate limiting* ed errori, acquisizione completa delle risorse Shopify, analisi statica del codice (Liquid, JSON, JS), *modello a grafo con livelli di confidenza*, *snapshot* immutabili con *diff* e motore di regole estensibile.
- **Accettazione:** criteri severi (*test pass* $\geq 95\%$, *falsi positivi ad alta severità* $\leq 10\%$), da verificare su uno store Shopify creato autonomamente dal gruppo.

2.1.2 Considerazioni e criticità

Il capitolato unisce requisiti stringenti a sfide ingegneristiche stimolanti: dallo sviluppo di *scansioni durevoli e ripristinabili* (con gestione di crash, *retry*, idempotenza e avanzamento) alla progettazione di un *modello a grafo delle dipendenze* per rilevare *codice morto* tramite analisi statica e API, fino alla sicurezza nell'esecuzione isolata di codice non fidato. A bilanciare queste opportunità intervengono però criticità: un supporto proponente limitato a incontri quindicinali e a un referente unico, senza alcun supporto infrastrutturale; la forte dipendenza dall'ecosistema Vercel/Supabase; la natura di nicchia della piattaforma Shopify.

2.1.3 Motivazioni dello scarto

Il fattore determinante è stato il carico organizzativo e operativo richiesto: il gruppo dovrebbe allestire in autonomia lo store di sviluppo usato come insieme di controllo per i *falsi positivi* e monitorare costantemente i limiti dei piani gratuiti di Vercel e Supabase. A ciò si aggiungono le criticità già descritte, che rendono l'opportunità di apprendimento poco spendibile al di fuori di questo specifico ecosistema.

2.2 C5 - RareCare Navigator (Miriade S.r.l.)

2.2.1 Obiettivo e tecnologie

RareCare Navigator è una piattaforma cross-platform con assistente AI progettata per la raccolta dati, il coordinamento e il supporto decisionale nelle malattie rare, a partire dalla Sindrome di Sanfilippo. Lo scopo principale è affiancare la figura umana del *Care Navigator* e ridurre il carico organizzativo delle famiglie, connettendo in un unico spazio sicuro medici, scuola, terapisti e servizi sociali.

Il sistema si fonda su un *Core* estendibile ad altre patologie tramite pacchetti di configurazione (*Disease Pack*) e prevede la validazione sul campo con l'associazione Sanfilippo Fighters.

Il livello di complessità tecnica del progetto è medio-alto e si articola sui seguenti aspetti:

- **Tecnologie consigliate (non vincolanti):** frontend in Flutter o React Native; backend *serverless* su AWS in regione UE (API Gateway, Lambda, Step Functions, EventBridge, IaC); database Aurora Serverless v2 (PostgreSQL con pgvector) e bucket S3 cifrati; servizi AI su Amazon Bedrock, Textract e Transcribe (o alternative OpenAI/Hugging Face).
- **Funzionalità chiave:** permessi granulari basati su ruolo, relazione con il paziente e attributi del documento (RBAC/ReBAC/ABAC), dashboard collaborativa e fascicolo digitale cifrato (referti, PAI, PEI), workflow per checklist dinamiche e scadenze con notifiche ed escalation, chatbot RAG basato esclusivamente su fonti verificate con citazioni e *Guaradrails*, pipeline NLP/NER per estrarre entità cliniche previa conferma umana obbligatoria e moduli opzionali (assistente vocale, mappa geospaziale e ML predittivo).
- **Organizzazione e conformità:** gestione di dati sanitari di minori (art. 9 GDPR) con cifratura TLS 1.3 e AES-256 (KMS), OAuth2/MFA (Cognito), registro di audit immutabile e accessibilità WCAG 2.1 AA; redazione della valutazione d'impatto DPIA, monitoraggio dei consumi per rientrare nel *free tier* AWS e mantenimento dell'AI entro un perimetro informativo per non ricadere nei vincoli sui dispositivi medici (MDR) e sull'EU AI Act.

L'opportunità di apprendimento risulta molto completa e trasversale, unendo progettazione DDD ed esagonale, sviluppo mobile/web, cloud serverless su AWS, sicurezza avanzata e GenAI in ambito clinico.

2.2.2 Considerazioni e criticità

La documentazione è estremamente chiara e include l'Allegato 1, una guida implementativa che definisce una roadmap incrementale, analizza i *trade-off* cloud e risolve preventivamente le criticità dei requisiti (come il bilanciamento tra cifratura *end-to-end* ed elaborazione AI lato server, l'estensione del modello RBAC e il mancato supporto dell'italiano nei servizi medicali AWS). Anche il supporto della proponente si attesta su un livello elevato, offrendo tre referenti dedicati (Cloud/AI, UX/UI e dominio clinico), formazione iniziale, sale riunioni a Padova e Thiene e piena flessibilità nel concordare un perimetro sostenibile per il gruppo.

A fronte di questi vantaggi, emergono tuttavia alcune criticità: l'elevato onere di analisi normativa e architetturale richiesto dalla tutela dei dati sanitari pediatrici (DPIA, consensi granulari, confini MDR/AI Act), l'assenza di *mockup* grafici dell'interfaccia e la mancata fornitura diretta di budget o account cloud aziendali, che lascia al gruppo l'onere di presidiare i limiti del piano gratuito di AWS.

2.2.3 Motivazioni dello scarto

Nonostante il forte impatto sociale, l'ottimo supporto multidisciplinare offerto da Miriade S.r.l. e l'interesse per lo stack serverless e GenAI su AWS, il gruppo ha deciso di tenere questo capitolato come opzione secondaria.

Nella scelta finale hanno pesato il considerevole carico normativo e progettuale richiesto dalla gestione dei dati sanitari e dallo sviluppo full-stack multiplatforma.

3 Altri capitoli valutati

3.1 C0 - AIRA (Sanmarco Informatica)

3.1.1 Obiettivo

Il progetto AIRA (AI Requests Automation) nasce con l'obiettivo di risolvere un problema molto comune nelle aziende: la dispersione delle informazioni su strumenti disorganizzati. Poiché l'eterogeneità dei canali di comunicazione e la mancanza di una centralizzazione causano continue

perdite di tempo per l'inserimento manuale di ordini e richieste, l'idea è sviluppare un software guidato dall'intelligenza artificiale per automatizzare l'intero flusso.

Il sistema monitora i canali di contatto applicando un filtro di sicurezza sui dati in ingresso. Successivamente, analizza i contenuti estraendo le informazioni in base al formato: interpreta i testi delle e-mail, trascrive le note vocali ed elabora immagini e allegati sfruttando tecnologie OCR e di visione artificiale.

Compresa la semantica della richiesta, i dati vengono convertiti in formati strutturati come JSON, classificati per tipologia e inoltrati automaticamente ai gestionali aziendali (B2B, ERP, CRM) tramite API REST, azzerando l'intervento umano. Viene integrata un'interfaccia amministrativa per configurare i canali attivi, gestire la persistenza dei dati e consultare lo storico dei log.

3.1.2 Motivazioni dello scarto

Il capitolato propone un uso interessante dell'intelligenza artificiale e si fa apprezzare per la chiarezza del modello architetturale e delle tecnologie consigliate. Tuttavia, è stato scartato perché risulta interamente focalizzato sull'automazione del lavoro d'ufficio e sull'elaborazione di documenti commerciali, uno scenario che non rientra tra i principali interessi del gruppo.

3.2 C2 - NEXUM Concierge (Eggon)

3.2.1 Obiettivo

Il progetto proposto da Eggon ha lo scopo di introdurre un assistente conversazionale, denominato Concierge, all'interno di NEXUM. Questa piattaforma è dedicata alla comunicazione aziendale e alla gestione dei servizi HR e mira a semplificare l'accesso alle informazioni e le attività più ricorrenti della quotidianità aziendale attraverso il solo utilizzo del linguaggio naturale. La proponente, per rendere realizzabile il progetto, circoscrive l'MVP (Minimum Viable Product) alla sola gestione di ferie e permessi.

Il sistema prevede la creazione di una chat testuale in grado di comprendere i messaggi e conservare le informazioni ricevute in precedenza, deducendo dal contesto i dati mancanti o chiedendo chiarimenti in caso di richieste ambigue. Il Concierge dovrà integrare diverse capacità: la consultazione della conoscenza aziendale (come i regolamenti interni), la lettura dei dati personali dell'utente (come le ferie richieste o residue) e la conoscenza del prodotto, garantita dall'integrazione dei manuali d'uso forniti da Eggon.

Infine, il Concierge dovrà essere in grado di agire operativamente, guidando l'utente alla compilazione del piano di ferie e assistendolo fornendogli un riepilogo sui dati inseriti e richiedendo conferma esplicita. A livello architetturale, questo sarà eseguito con *Gate applicativi* che garantiscono il rispetto delle autorizzazioni di accesso e la tracciabilità delle operazioni.

3.2.2 Motivazioni dello scarto

Nonostante il capitolato affronti un problema aziendale concreto, la proposta è stata scartata poiché il gruppo ha preferito orientarsi verso altre tematiche, con l'obiettivo di misurarsi con domini applicativi differenti e maggiormente allineati agli interessi del gruppo.

3.3 C3 - Fossdot (Uranion EMEA Ltd)

3.3.1 Obiettivo

Il progetto, proposto da Uranion EMEA Ltd con il nome FossDot, punta a sviluppare un ecosistema open source completo per uno *smart speaker* vocale che sia estensibile e attento alla privacy, pensato come alternativa ai classici dispositivi chiusi (tipo Amazon Echo Dot). Il sistema copre tutta la catena del prodotto: dalla scelta dell'hardware di riferimento al *firmware*

embedded per catturare l'audio e rilevare la *wake word*, passando per un servizio di *bridging* che collega il dispositivo a provider di AI tramite una pipeline STT (*speech-to-text*), LLM e TTS (*text-to-speech*), fino a un'interfaccia web utile a registrare e gestire più dispositivi. Il tutto deve essere rilasciato con licenza permissiva (Apache 2.0 o MIT) e testato end-to-end su un dispositivo fisico.

3.3.2 Motivazioni dello scarto

Il capitolato ci è sembrato molto interessante (uno dei più stimolanti in assoluto), ma di difficoltà eccessiva per il gruppo. Il progetto richiede infatti di mettere le mani sull'hardware vero e proprio, che comprende scelta e validazione di scheda, array di microfoni e altoparlante e realizzazione di un prototipo funzionante, oltre a scrivere il firmware embedded, per cui viene consigliato ed è preferenziale l'uso di Rust. Il gruppo non ha né competenze pratiche di elettronica né una formazione ingegneristica adatta a questo tipo di lavoro. A tutto ciò si aggiunge anche l'ampiezza del capitolato, che si articola in quattro componenti distinte che devono poi integrarsi perfettamente in un collaudo end-to-end su dispositivo fisico, rendendo poco realistico il completamento dell'MVP nei tempi previsti.

3.4 C4 - Piattaforma per la gestione dei veicoli e dei servizi post-vendita (M31 S.r.l.)

3.4.1 Obiettivo

Il progetto proposto da M31 S.r.l. ha come obiettivo lo sviluppo di una piattaforma software utile alla gestione dell'intero ciclo di vita dei veicoli e dei servizi post-vendita. Lo scopo finale risiede nella centralizzazione delle informazioni, mettendo in comunicazione la casa automobilistica, le concessionarie e i proprietari finali dei mezzi.

L'architettura logica è suddivisa in tre livelli, ciascuno con responsabilità e accessi indipendenti. Sarà possibile la comunicazione tra i servizi sia in modo sincrono, con l'utilizzo di API, che asincrono, tramite lo scambio di eventi.

La casa automobilistica gestisce il catalogo dei modelli e organizza le campagne di richiamo; avrà inoltre la possibilità di registrare i singoli veicoli (tramite identificativo univoco) e associarli alle concessionarie. Le concessionarie si occupano della vendita, dell'associazione dei veicoli ai clienti e delle richieste di manutenzione. Infine, i proprietari hanno accesso a un'area personale da cui possono consultare le statistiche del mezzo (come chilometraggio e stato della batteria, inserite manualmente o tramite simulatore) e prenotare appuntamenti.

A supporto di queste funzionalità, è richiesta un'architettura che preveda l'impiego di Node.js e Go per i servizi applicativi, l'utilizzo di database relazionali e non relazionali e la creazione di interfacce web in Angular. Tutto ciò dovrà essere racchiuso all'interno di un'infrastruttura gestita tramite Kubernetes.

3.4.2 Motivazioni dello scarto

Nonostante il capitolato presenti un'architettura stimolante, la proposta è stata scartata poiché il dominio applicativo legato al settore automobilistico non ha suscitato un forte interesse all'interno del gruppo. Il team ha infatti preferito orientarsi verso altre tematiche e valutare capitolati ritenuti più affini ai gusti personali.

3.5 C6 - Sistema AI di caching semantico e dashboard FinOps (Var Group)

3.5.1 Obiettivo

Il progetto proposto da Var Group nasce con l'obiettivo di risolvere due criticità molto diffuse nell'adozione aziendale dell'intelligenza artificiale generativa su larga scala: la ridondanza delle

richieste inviate ai modelli e la mancanza di visibilità immediata sui costi di inferenza a consumo, spesso rilevati solo a fine mese con la fattura dei vari provider.

L'idea consiste nell'estendere LiteLLM, un orchestratore e proxy open source già utilizzato in produzione dall'azienda per instradare le chiamate verso molteplici LLM, integrandovi due nuovi componenti originali.

Il primo modulo introduce un livello di *caching semantico* basato su *embedding* e ricerca di similarità (*cosine similarity*). A differenza delle cache tradizionali, che richiedono una corrispondenza esatta del testo, questo sistema riconosce domande formulate in modo diverso ma dal significato equivalente e restituisce direttamente la risposta salvata su Redis o database vettoriale (Qdrant o pgvector), registrando i token e la latenza risparmiati.

Il secondo componente consiste in una dashboard FinOps *multi-tenant*, realizzata come servizio separato con aggiornamento quasi in tempo reale. L'interfaccia aggrega e isola i consumi per team, progetto e singolo utente, quantifica il risparmio effettivo generato dal *caching semantico* e permette di configurare budget con avvisi automatici e previsioni (*forecasting*) della spesa mensile.

3.5.2 Motivazioni dello scarto

Il capitolato affronta una problematica aziendale concreta ed è stato apprezzato per la chiarezza della documentazione, la solidità dello stack tecnologico consigliato e il piano di formazione strutturato offerto dalla proponente.

Tuttavia, è stato scartato perché fortemente incentrato sulla rendicontazione economica dei consumi (FinOps) e sull'estensione di un proxy preesistente, un dominio applicativo che ha suscitato minore interesse nel gruppo rispetto a soluzioni più tangibili, a cui si aggiunge l'onere procedurale richiesto prima dell'implementazione.

3.6 C7 - SmartDashboard (Ergon Informatica S.r.l.)

3.6.1 Obiettivo

Il progetto SmartDashboard mira a realizzare una piattaforma di *Business Intelligence* end-to-end, pensata per democratizzare l'accesso ai dati aziendali. Poiché le informazioni risiedono spesso in sistemi ERP complessi e richiedono interventi tecnici per la reportistica, l'obiettivo è costruire un sistema *self-service* che consenta agli utenti di esplorare autonomamente i dati.

L'architettura prevede l'estrazione dei dati grezzi dal gestionale, la loro trasformazione (ETL) e il consolidamento in un *data warehouse* ottimizzato. Su questa base si innesterà una web application dotata di interfaccia drag & drop, permettendo la composizione personalizzata di dashboard con widget, grafici, mappe e indicatori di sintesi.

L'elemento distintivo è l'integrazione di un layer di intelligenza artificiale conversazionale. Questo assistente permetterà di generare visualizzazioni a partire da richieste in linguaggio naturale e analizzerà in automatico i dati per evidenziare anomalie, scostamenti e tendenze, fornendo spiegazioni chiare e immediate senza richiedere la conoscenza del modello dati sottostante.

3.6.2 Motivazioni dello scarto

Nonostante l'attrattiva di sviluppare un'applicazione web completa e la stimolante integrazione dell'assistente virtuale, il capitolato non è stato selezionato. La decisione è derivata dalla marcata focalizzazione su processi di *data engineering* e pipeline ETL, ambiti analitici che si discostano dai principali interessi di sviluppo software del gruppo.

3.7 C9 - SpecCheck (Zucchetti S.P.A.)

3.7.1 Obiettivo

Il progetto SpecCheck si pone l'obiettivo di affrontare il problema della verifica del software generato tramite intelligenza artificiale, garantendo la tracciabilità end-to-end. L'idea centrale è quella di sviluppare estensioni, denominate *skills*, per la piattaforma OpenCode, in modo da automatizzare il controllo di coerenza tra le varie fasi del ciclo di vita del software.

Il sistema proposto dovrà leggere e interpretare i diagrammi dei casi d'uso e la relativa documentazione, per poi confrontarli con le specifiche dei requisiti. Successivamente, dovrà analizzare il codice sorgente prodotto per comprenderne il comportamento effettivo e verificare che questo implementi fedelmente quanto definito nei requisiti iniziali.

Infine, il progetto prevede la realizzazione di una dashboard capace di evidenziare lo stato di avanzamento e segnalare tempestivamente al programmatore eventuali discrepanze. Le estensioni create andranno a integrarsi nel sistema di *harness* di OpenCode (tramite skill, tool e plugin), fungendo da strumento di validazione automatico.

3.7.2 Motivazioni dello scarto

Il capitolato presenta una tematica innovativa, affrontando il problema attuale della validazione del codice prodotto da LLM. Tuttavia, è stato scartato per la sua natura marcatamente esplorativa e per lo stretto vincolo con l'ecosistema OpenCode, limitando la libertà del gruppo nella scelta e progettazione dello stack tecnologico.