A.W.A.R.E: An AI Agent for Water Autonomy, Resilience, and Efficiency

Raymond Li & Jack Liang
Master's in Software Engineering
San Jose State University
San Jose, United States
raymond.li01@sjsu.edu
jack.liang@sjsu.edu

Sophia Atendido & Dhruv Verma
Master's in Software Engineering
San Jose State University
San Jose, United States
sophia.atendido@sjsu.edu
dhruv.verma01@sjsu.edu

***Abstract*—Aging distribution assets, non-revenue water (NRW), volatile electricity tariffs, and alarm fatigue from dense sensor deployments increasingly challenge municipal water utilities. Conventional SCADA platforms surface data but provide limited decision support, leaving operators to triage alarms and plan valve isolations under time pressure. This paper presents *A.W.A.R.E* (Agent for Water Autonomy, Resilience, and Efficiency): a proactive, self-healing AI agent that couples a digital twin of the water network with a multi-agent decision layer. A.W.A.R.E fuses acoustic, pressure, and flow telemetry into confidence-scored leak signals; simulates alternative valve closure sequences; and co-optimizes pump and tank schedules against time-varying tariffs while preserving hydraulic safety and regulatory constraints.**

The solution was designed using a Design Thinking approach grounded in interviews, empathy mapping, and journey mapping across operators, field technicians, asset managers, and residents. We detail the research insights, architecture, algorithms, security model, and test strategy, and propose success metrics spanning NRW, outage minutes, energy cost per million gallons, and operator trust. For practitioners and researchers, the paper positions A.W.A.R.E as a blueprint for integrating digital twins and multi-agent AI into critical water infrastructure in a safe, explainable, and human-centered manner.

***Index Terms*—Water distribution, digital twin, EPANET/WNTR, multi-agent systems, leak detection, sensor fusion, pump scheduling, SCADA, human-in-the-loop AI, explainability, ICS security, IEC 62443, NIST SP 800-82.**

I. Introduction

Water distribution utilities worldwide are under pressure from aging infrastructure, constrained capital budgets, and climate-induced variability in supply and demand. Pipe bursts, background leaks, and unauthorized consumption contribute to non-revenue water (NRW), which can exceed 30% in some systems [3], [4]. At the same time, utilities face volatile electricity tariffs and peak demand charges that amplify operating costs when pumps run during on-peak periods.

Digitalization has brought low-cost sensors, advanced metering infrastructure (AMI), and SCADA systems that generate high-frequency telemetry and alarms. However, many utilities report that this data is underutilized. Operators describe "Christmas tree" events when dozens of alarms arrive in minutes, many of which are mutually redundant or spurious. In such conditions, humans must mentally integrate telemetry, GIS maps, and valve atlases while under time pressure, which increases the probability of suboptimal decisions and extended outages.

A. Challenges in Water Utility Operations

Key operational challenges highlighted by industry guidance and field interviews include:

- **NRW and leak management:** Identifying and localizing leaks, estimating their severity, and prioritizing repairs in a resource-constrained environment [3], [7].
- **Isolation planning:** Designing valve closure sequences that safely isolate a failure while minimizing customer impact and preserving pressure floors.
- **Energy-aware pumping:** Coordinating pumps and tanks to exploit off-peak tariffs without compromising service levels or asset health [10].
- **Alarm fatigue:** Managing dense alarm streams from SCADA and sensor networks without overwhelming operators.
- **Trustworthy autonomy:** Introducing AI-based decision support in safety-critical environments while preserving transparency, human authority, and regulatory compliance.

B. Role of AI and Digital Twins

Digital twins—computational replicas of physical systems—have been applied in water distribution to support scenario analysis, planning, and real-time decision support [5], [6], [11]. When coupled with AI, digital twins can enable:

- *Model-based leak localization* via residuals between observed and simulated states;
- *Simulation-driven planning* for isolation sequences and pump schedules;
- *What-if analysis* for proposed operational changes or capital interventions.

Multi-agent systems provide a natural way to modularize complex decision-making, with specialized agents for leak detection, isolation planning, and energy optimization, coordinated through shared state and constraints.

C. Objectives and Contributions

A.W.A.R.E (Agent for Water Autonomy, Resilience, and Efficiency) is a prototype platform that combines:

- A high-fidelity, versioned digital twin of the distribution network;
- A multi-agent AI layer for leak detection, isolation, and energy optimization;
- An operator-centric web UI with explainability and auditability;
- A security architecture aligned with ICS best practices [12], [13].

The main contributions of this paper are:

1) A Design Thinking-driven requirements analysis for AI decision support in water utilities;
2) A modular architecture that integrates React/TypeScript, FastAPI, Supabase, and EPANET/WNTR-compatible digital twins;
3) A sensor fusion formulation for leak likelihood and a scoring framework for valve isolation and energy optimization plans;
4) A security, testing, and evaluation plan suitable for critical infrastructure deployments.

D. Paper Organization

Section II reviews background and related work. Section III presents the Design Thinking process. Section IV details the system architecture. Section V describes the data model and digital twin. Section VI explains the multi-agent AI layer. Section VII outlines core functionalities and use cases. Section VIII discusses personas and UX. Section IX covers deployment and operations. Section X presents the security model. Section XI describes testing and verification. Section XII proposes evaluation metrics, followed by discussion and conclusion.

II. Background and Related Work

A. Non-Revenue Water and Leak Management

NRW encompasses real losses (leaks, bursts), apparent losses (meter inaccuracies, theft), and unbilled authorized consumption [3]. Guidance from AWWA and IWA emphasizes systematic water audits, pressure management, active leak detection, and speed of repair [3], [4]. Traditional leak management relies on:

- *District Metered Areas (DMAs):* Partitioning networks into DMAs to monitor night flows and detect anomalies;
- *Acoustic methods:* Correlators and loggers that detect leak noise in pipes;
- *Step tests:* Sequential valve closures to localize leaks.

Model-based methods use hydraulic simulators to compute expected pressures/flows and compare them against observations [7]. Data-driven methods apply machine learning to AMI and sensor data to detect anomalies [8], [9].

B. Digital Twins for Water Networks

Water network simulators such as EPANET [5] and WNTR [6] support hydraulic and water quality analysis. Modern practice increasingly uses digital twins that:

- Integrate real-time telemetry and SCADA signals;
- Maintain configuration versions for pipes, valves, and tanks;
- Support online calibration and parameter estimation;
- Enable scenario analysis for resilience planning [11].

A key challenge is aligning the twin with field reality given GIS/SCADA drift and incomplete valve status information.

C. Multi-Agent Systems and Explainable AI

Multi-agent systems decompose complex decision-making into specialized agents that coordinate through shared state and protocols. For critical infrastructure, AI assistants must be explainable and auditable to gain operator trust. Techniques such as local feature attributions, rule-based summaries, and counterfactuals can help explain predictions and recommendations while remaining actionable.

D. Industrial Control System Security

Industrial control systems (ICS) operate under different threat and risk profiles than traditional IT. NIST SP 800-82 [12] and IEC 62443 [13] stress network segmentation, defense-in-depth, least privilege, and rigorous change management. Any AI component must integrate into this security posture, avoid introducing new attack surfaces, and support traceability and rollback.

III. Design Thinking and User Research

We adopted a Design Thinking framework with phases of *Empathize*, *Define*, and *Ideate* prior to implementation [2]. This ensured that AI and automation were grounded in real operator workflows and constraints rather than purely technological possibilities.

A. Empathize: Research Goals and Methods

The primary goal was to understand how leak and pressure events are handled in live operations, what tools and workarounds are used, and what barriers exist to adopting AI decision support.

1) Participants: Stakeholders included:

- **Control room operators** (primary users);
- **Field technicians** responsible for physical valve operations and repairs;
- **Asset managers** overseeing long-term performance and budgets;
- **SCADA engineers** responsible for telemetry and control systems;
- **Secondary stakeholders** such as sustainability officers, finance analysts, and public works leadership.

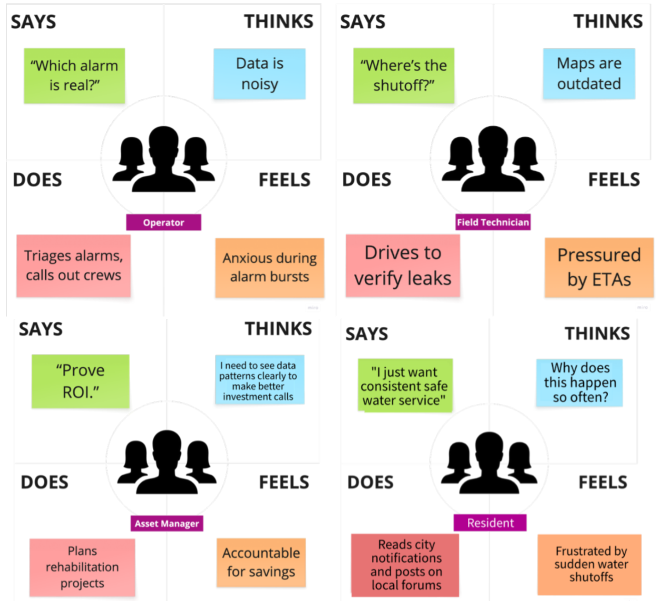

Fig. 1. Empathy maps for operator, field technician, asset manager, and resident personas.

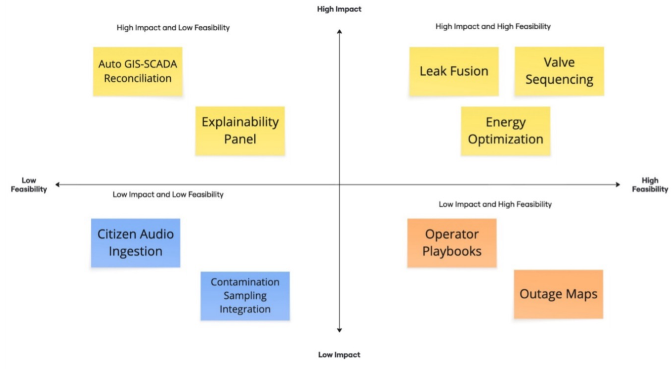

Fig. 2. Concept portfolio across feasibility and impact; selected concepts occupy the high-feasibility, high-impact quadrant.

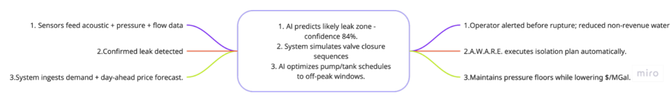

Fig. 3. End-to-end loop: sensors produce acoustic/pressure/flow data; AI predicts likely leak zones, simulates valve sequences and energy schedules; operators review and act.

2) Methods: We used multiple qualitative and quantitative methods [2]:

- *Semi-structured interviews* (45–60 minutes) focusing on recent leak events, alarm handling, and energy decisions.
- *Contextual inquiry and shadowing* during two alarm bursts, observing tools, communication patterns, and workarounds.
- *Artifact analysis* of incident reports, alarm logs, energy bills, leak tickets, and valve maintenance records.
- *Surveys* capturing perceived alarm fatigue, response times, and satisfaction with existing tooling.

B. Empathy Maps and Leak Journey

Empathy maps were synthesized for four representative personas: operator, field technician, asset manager, and resident (Fig. 1). Each map captured what the persona *says*, *thinks*, *does*, and *feels* during leak and pressure events.

A consolidated journey map for a typical leak identified eight stages: (1) alarm burst, (2) SCADA/GIS visual check, (3) dispatch, (4) on-site verification, (5) valve isolation, (6) pressure rebalancing, (7) customer communication, and (8) postmortem. Pain points included:

- Noisy and duplicative alarms with inconsistent severity;
- Difficulty locating accurate valve maps and confirming valve status;
- Limited reuse of incident learnings for future events.

C. Define: Problem Statement and "How Might We"

A synthesized point-of-view statement was: *Night-shift operators need high-confidence, explainable leak pre-emption and isolation guidance, because alarm floods and outdated maps delay decisive action and increase water loss and outage minutes.*

This led to "How might we" questions [1], [2]:

1) Fuse acoustic, pressure, and flow into a single actionable leak signal.
2) Recommend or execute valve sequences that minimize customer impact.
3) Co-optimize pump and tank schedules against day-ahead prices while protecting pressure floors.
4) Provide explanations and "why now" rationales for each recommendation.
5) Keep the digital twin aligned with field reality as the system evolves.

D. Ideate: Concept Portfolio and Selection

Brainstorming techniques (Crazy-8s, SCAMPER, and "worst possible idea") generated over fifty concepts, clustered into six themes: sensing & fusion, isolation UX, energy operations, trust, data quality, and civic signals. Each concept was scored along feasibility and impact (Fig. 2); high-feasibility, high-impact ideas formed the initial scope.

The resulting system loop (Fig. 3) proceeds from sensing to AI inference to simulation and operator action.

IV. System Architecture

The A.W.A.R.E prototype adopts a modular, service-oriented architecture that separates concerns between presentation, orchestration, data, and analytics layers. This section describes the logical view, technology stack, data flows, and scalability considerations.

A. Logical View

At a high level, the system consists of:

- A **React/TypeScript frontend** for operator dashboards, simulation tools, and administrative panels.

- A **FastAPI backend** that exposes RESTful endpoints for telemetry ingestion, simulations, and audit logs.
- A **Supabase/PostgreSQL data layer** providing persistence, authentication, RBAC, and real-time change streams.
- A **digital twin and analytics engine** that encapsulates hydraulic simulations and AI agents.

B. Frontend (React + TypeScript + Vite)

The frontend is built using React and TypeScript, bundled via Vite for fast development and optimized builds. Key UI surfaces include:

- *Operations dashboard:* Real-time pressure, flow, and acoustic signals; leak heatmaps; alarm cards; and incident timelines.
- *Isolation planner:* Map-centric interface for inspecting candidate valve sequences, visualizing affected zones, and approving plans.
- *Energy console:* Baseline and optimized pump schedules, tank trajectories, and tariff overlays.
- *Explainability panels:* Narrative explanations, key features, and supporting evidence for each recommendation.
- *Admin views:* Role management, configuration of thresholds, and simulation settings.

Supabase Realtime channels stream updates (e.g., new recommendations, updated incidents) to the frontend, enabling live dashboards without manual refresh.

C. Backend (Python FastAPI)

The backend orchestrates data ingestion, analytics, and simulation:

- *Telemetry ingestion services* validate and persist sensor data from SCADA/IoT gateways.
- *Analytics services* compute leak likelihoods, isolation rankings, and energy schedules using the digital twin.
- *Notification services* push updates to the frontend via Supabase channels.

FastAPI was chosen for its asynchronous support, automatic OpenAPI documentation, and strong typing.

D. Data Layer and Digital Twin (Supabase/PostgreSQL)

Supabase provides:

- Postgres as the relational database;
- Authentication and RBAC primitives;
- Real-time subscriptions on table changes.

Schema design includes tables for network topology (nodes, pipes, valves, tanks), time-series telemetry, incidents, recommendations, user roles, and audit logs. Topology tables align with EPANET/WNTR constructs [5], [6], [11].

E. Scalability and Fault Tolerance

The architecture is designed to scale horizontally:

- Stateless FastAPI instances behind a load balancer;
- Background workers for long-running simulations;
- Database connection pooling and read replicas for reporting workloads.

Fault tolerance is supported through retry policies on telemetry ingestion, circuit breakers around external connectors, and graceful degradation in the UI when components are temporarily unavailable.

V. Data Model and Digital Twin Implementation

The digital twin is central to A.W.A.R.E, enabling simulation-driven decision support.

A. Topology and Asset Modeling

The physical network is represented using:

- **Nodes:** Junctions with elevations and base demands;
- **Links:** Pipes, pumps, and valves with diameters, roughness, and status;
- **Tanks/Reservoirs:** Storage and supply boundaries with level constraints;
- **Zones:** Logical groupings (e.g., pressure zones, DMAs).

Tables in Postgres encode these entities and their attributes. A translation layer exports network snapshots into EPANET or WNTR input formats for simulation [5], [6].

B. Telemetry and Event Modeling

Telemetry from sensors includes:

- Pressure at selected nodes;
- Flow in key mains and DMAs;
- Acoustic intensity or leak noise metrics.

Each measurement is stored with timestamp, location, quality flags, and source metadata. Events such as *alarms*, *recommendations*, and *operator actions* are normalized into a common event table to support auditability and replay.

C. Calibration and Model Management

To remain useful, the twin must be calibrated against observed data:

- *Demand calibration:* Adjusting base demands and patterns to match observed flows.
- *Roughness calibration:* Fitting pipe roughness to reproduce observed pressures.
- *Valve status reconciliation:* Incorporating technician confirmations to reduce uncertainty.

Configuration versions are managed so that simulations can reference a specific model snapshot; incident replay uses the model version active at the incident time.

VI. Multi-Agent AI and Analytics

A.W.A.R.E employs a multi-agent design where specialized agents collaborate via shared state and a coordination layer.

A. Leak Detection and Sensor Fusion

Consider a sliding time window of telemetry aggregated into a feature vector $x_t \in \mathbb{R}^d$ composed of:

- Pressure residuals between measured and simulated pressures at key nodes;
- Flow anomalies, e.g., deviations from expected DMA night flows;

- Acoustic indicators such as spectral energy in leak-sensitive bands;
- Contextual features (pipe age, material, historical leak density).

A calibrated probabilistic model yields segment-level leak probabilities:

$$p(\text{leak}_i \mid x_t) = \sigma\big(\beta_{0,i} + \beta_i^\top x_t\big), \tag{1}$$

where $\sigma(\cdot)$ is the logistic function and parameters $(\beta_{0,i}, \beta_i)$ can differ by zone or pipe class. Spatial smoothing incorporates adjacency and hydraulic proximity, reflecting that neighboring segments may be jointly affected [7]–[9].

B. Valve Isolation Planning

The isolation agent operates once a segment or small region exceeds a leak-likelihood threshold. Given a candidate failure set $\mathcal{F}$ and the valve set $\mathcal{V}$, the agent:

1) Enumerates feasible valve subsets $\mathcal{S} \subseteq \mathcal{V}$ that isolate $\mathcal{F}$.
2) For each $\mathcal{S}$, updates the twin (closing valves in $\mathcal{S}$) and simulates steady-state hydraulics.
3) Computes a score function

$$J(\mathcal{S}) = w_1 C(\mathcal{S}) + w_2 O(\mathcal{S}) + w_3 R(\mathcal{S}), \tag{2}$$

where C is customer impact (e.g., count of low-pressure nodes), O is operational risk/complexity, and R captures resilience (e.g., storage and redundancy).
4) Ranks candidates and returns the top-k sequences with predicted impacts.

Because exhaustive enumeration may be combinatorial, heuristics prune valve sets based on topological distance and previous incidents [11].

C. Energy Optimization

The energy agent solves a constrained optimization problem over a time horizon T (e.g., 24 hours) divided into discrete intervals:

$$\begin{aligned}
\min_{\{u_t\}} \quad & \sum_{t=1}^{T} c_t\, E(u_t) \qquad (3)\\
\text{s.t.} \quad & h_i(t) \geq h_i^{\min}, \quad \forall i, t,\\
& s^{\min} \leq s(t) \leq s^{\max}, \quad \forall t,\\
& s(t+1) = s(t) + f(u_t, d_t), \quad \forall t,\\
& u_t \in \mathcal{U}, \quad \forall t,
\end{aligned}$$

where c_t is the tariff, $E(u_t)$ is energy consumption, $h_i(t)$ is head at node i, $s(t)$ is tank storage, d_t is demand, and $\mathcal{U}$ captures pump and valve control bounds [10]. The current prototype uses mixed-integer linear or quadratic approximations sufficient for near-real-time scheduling, with the twin used to validate candidate schedules.

D. Coordination and Explainability

A coordination layer enforces consistency between agents. For example, the energy agent must respect active isolation plans, and isolation plans must consider energy implications. Explainability is provided by:

- Feature-level attributions for leak predictions;
- Decompositions of $J(\mathcal{S})$ into customer, operational, and resilience components;
- Comparative views (baseline vs. recommended pump schedules).

These explanations are surfaced in the UI as both visual and textual narratives.

VII. Core Functionalities and Use Cases

A. Leak Detection and Signal Fusion

In operations, A.W.A.R.E continuously ingests telemetry and updates leak-likelihood maps. Operators can:

- View ranked leak candidates on a map, color-coded by probability;
- Inspect time-series plots and residuals that support each candidate;
- Adjust sensitivity thresholds per zone.

For deeper reading on leak detection methods, see [7]–[9].

B. Autonomous Valve Isolation Sequencing

When a leak candidate surpasses a configurable threshold, the system automatically triggers the isolation agent. The UI presents:

- A list of candidate sequences with scores and key metrics (customers affected, low-pressure nodes, estimated outage minutes);
- A step-by-step view of each sequence;
- A side-by-side comparison of the top two or three options.

1) Use Case: Nighttime DMA Leak: In a nighttime scenario, low demand amplifies the hydraulic signal of a leak. A.W.A.R.E flags a segment with high leak likelihood, proposes three sequences, and shows that the top-ranked plan isolates the leak with minimal service impact. The operator approves the plan and dispatches technicians with mobile instructions.

C. Dynamic Energy Optimization

The energy agent computes day-ahead pump schedules based on tariff curves and demand forecasts. Asset managers and operators can:

- Compare baseline vs. optimized cost and headroom;
- Configure risk preferences (e.g., minimum storage buffers);
- Lock-in certain pumps or time windows for operational or contractual reasons.

For additional background on water-energy optimization, see [10], [11].

D. Explainability and Trust Layer

Each recommendation includes:

- Key drivers (e.g., pressure drop at a specific node, anomalous DMA flow);
- Confidence intervals and expected benefit ranges;
- A "why not" explanation for alternative plans not selected as top-ranked.

A dry-run mode enables operators to execute recommendations on a sandboxed digital twin without affecting live systems, aiding training and trust-building.

E. System Monitoring and Reporting

The platform provides:

- Real-time health dashboards for the system itself (latency, error rates, backlog of simulations);
- Historical incident timelines and operator actions;
- KPI reports on NRW, outage minutes, and energy cost per MGal for planning and regulatory reporting.

VIII. Personas and User Experience

A.W.A.R.E is designed for multiple user groups with overlapping but distinct needs.

A. Control Room Operators

Operators require:

- A low-noise, high-signal view of alarms and anomalies;
- Rapid access to isolation and energy plans;
- Clear indications when AI is recommending actions vs. when actions are purely human-driven.

The UI emphasizes consistency with existing SCADA screens, minimizing cognitive load while introducing new capabilities.

B. Field Technicians

Technicians access mobile-friendly screens showing:

- The target leak area on a map;
- Ordered valve closure steps with clear IDs and photos;
- Ability to confirm or correct valve status, uploading photos that feed back into the twin.

C. Asset Managers and SCADA Engineers

Asset managers track trends in NRW, outage minutes, and energy spend. SCADA engineers use the system's API-first design and logs to:

- Integrate new sensors and telemetry streams;
- Calibrate and validate models;
- Investigate issues via audit trails.

D. Residents and Community Stakeholders

While residents are not direct users, roadmap features include:

- Outage notification APIs that utilities can connect to existing communication channels;
- Reliability dashboards that visualize improvements in leak response and service continuity.

TABLE I
RBAC Summary (Least Privilege)

Role	**Key Capabilities (examples)**
Operator	Approve isolation, trigger simulations, annotate incidents
Technician	Execute valve steps, confirm valves, upload photos
Asset Manager	Review KPIs, energy plans, set policy constraints
SCADA Engineer	Integrate data sources, calibration, connectors
Admin	RBAC policies, tenant configs, audit export

IX. Deployment and Operations

A.W.A.R.E is designed to be deployable in on-premises data centers or secure cloud environments.

A. Environment Separation

Typical environments include:

- **Development:** Rapid iteration with synthetic/demo data;
- **Staging:** Integration with non-production SCADA feeds and realistic scenarios;
- **Production:** Hardened deployment with limited integration points and change windows.

B. CI/CD and Configuration Management

Containerized services (e.g., Docker) and infrastructure-as-code (e.g., Terraform) support reproducible deployments. CI/CD pipelines:

- Run unit and integration tests on each commit;
- Enforce code quality and security scans;
- Require approvals for production deployments.

Configuration (thresholds, tariff curves, model versions) is stored centrally and versioned to support rollback.

C. Observability

Metrics, logs, and traces are collected to support:

- Performance tuning (e.g., simulation latency);
- Capacity planning (e.g., telemetry ingestion load);
- Incident response (e.g., tracing a recommendation back through its computations).

X. API Security and Governance

Given the safety-critical nature of water infrastructure, the security architecture follows ICS best practices.

A. Identity, Session Management, and RBAC

Supabase Auth manages identity using secure credential storage and token-based sessions. Each frontend request includes a token that the backend validates. RBAC policies:

- Distinguish operators, technicians, asset managers, SCADA engineers, and administrators;
- Apply row-level security (RLS) policies so users see only permitted data;
- Support tenant-level isolation in multi-utility deployments [14], [15].

B. Defense-in-Depth Controls

Aligned with NIST SP 800-82 and IEC 62443 [12], [13], defense-in-depth includes:

- Network segmentation between OT (SCADA) and IT (analytics);
- TLS encryption for all API calls;
- Secrets management and rotation for API keys and credentials;
- Principle of least privilege for database and service accounts;
- Immutable audit logs with tamper-evident storage.

C. Audit Logging and Compliance

All recommendations, approvals, overrides, and configuration changes are logged with timestamps and user identities. Logs support:

- Event replay for forensics;
- Compliance reporting;
- Model behavior analysis (e.g., drift in prediction patterns).

XI. Testing and Verification

Testing spans frontend, backend, and digital-twin simulations.

A. Frontend Testing

Jest and React Testing Library validate:

- Component rendering and state transitions;
- User flows such as acknowledging alarms and approving sequences;
- Behavior under simulated API failures or latency.

Snapshot tests protect against accidental regressions in critical dashboards.

B. Backend Testing

Pytest-based tests cover:

- Telemetry validation and normalization logic;
- Leak-likelihood computations and boundary conditions;
- Isolation ranking and energy optimization for known scenarios.

Integration tests execute synthetic leak and pump events end-to-end using WNTR test networks [5], [6].

C. Performance and Resilience Testing

Performance testing evaluates:

- Maximum sustained telemetry rate;
- Latency for producing recommendations under load;
- Scalability of simulation workloads.

Resilience tests inject faults (e.g., delayed telemetry, failed simulations) to verify that the system degrades gracefully and surfaces actionable error messages to operators.

XII. Evaluation Metrics and Experiment Design

We propose a combination of leading indicators and lagging outcomes.

A. Leading Indicators

- **False-alarm rate:** Fraction of high-priority alerts that do not correspond to physical leaks (target $\leq 5\%$).
- **Time-to-first-action:** Time between detection and the first operator action (target ≤ 3 minutes).
- **Calibration error:** Deviation between predicted and observed leak frequencies.

B. Lagging Outcomes

- **NRW reduction:** Percentage reduction in NRW (target 10–20%) [3], [4].
- **Outage minutes:** Reduction in customer outage minutes (target 25–40%).
- **Energy cost:** Reduction in energy cost per MGal (target 12–25%).

C. Trust and Safety

- Percentage of operators opting into auto-execute modes (target $\geq 70\%$ by pilot end);
- Count of safety incidents attributable to the system (target: zero).

D. Experiment Design

A full-scale field trial would compare:

- Historical baseline performance vs. performance during an A.W.A.R.E pilot;
- Control zones using existing tools vs. treatment zones using A.W.A.R.E recommendations;
- Operator feedback collected via surveys and structured debriefs after major incidents.

XIII. Discussion and Future Work

The current A.W.A.R.E prototype demonstrates the feasibility of combining multi-agent AI, digital twins, and human-in-the-loop UX for water utilities. Several limitations remain:

- **Data realism:** The prototype relies on simulated or de-identified telemetry; live deployments will require extensive validation and contingency planning.
- **Model robustness:** Additional work is needed to handle sensor drift, topology inaccuracies, and data gaps.
- **Organizational adoption:** Successful deployment depends as much on training, culture, and governance as on technical capabilities.

Future work includes:

- Integrating with production SCADA and GIS systems via secure connectors;
- Extending the digital twin to multi-utility and city-scale networks;
- Exploring AR-assisted field tools for valve localization and confirmation;
- Incorporating citizen audio or image reports as optional corroborating signals under strict privacy and governance constraints;
- Applying transfer learning to share leak signatures and optimization policies across districts while respecting data sovereignty [1].

XIV. Conclusion

This paper presented *A.W.A.R.E*, an AI-driven agent for water autonomy, resilience, and efficiency. Grounded in Design Thinking research with operators, technicians, asset managers, and residents, the system fuses multimodal telemetry with a digital twin and multi-agent analytics to provide explainable leak isolation and energy optimization recommendations. By targeting reductions in NRW, outage minutes, and energy cost per million gallons while preserving strong human-in-the-loop controls and ICS-grade security, A.W.A.R.E offers a practical blueprint for trustworthy autonomy in critical water infrastructure.

Acknowledgment

The authors thank Prof. Rakesh Ranjan for his guidance and constructive feedback throughout the project, as well as peers who offered critiques during design reviews. They also acknowledge the open-source communities behind React, FastAPI, Supabase, Postgres, EPANET, WNTR, and related AI frameworks that enabled rapid development of the prototype.

References

[1] A.W.A.R.E Project Team, "A.W.A.R.E: Agent for Water Autonomy, Resilience, and Efficiency," CMPE 272 Project Report, San Jose State University, 2025.

[2] A.W.A.R.E Project Team, "A.W.A.R.E Design Thinking Report," CMPE 272, San Jose State University, 2025.

[3] American Water Works Association, *M36: Water Audits and Loss Control Programs*, 4th ed. Denver, CO: AWWA, 2016.

[4] M. Farley and R. Liemberger, "Developing a Non-Revenue Water Reduction Strategy—Part 1: Investigating and Assessing NRW," *Water21*, Int'l Water Association, 2005.

[5] L. A. Rossman, *EPANET 2 Users Manual*. U.S. Environmental Protection Agency, EPA/600/R-00/057, 2000.

[6] K. A. Klise, M. L. Bynum, R. Murray, and T. Haxton, "Water Network Tool for Resilience (WNTR)," U.S. EPA, 2017. [Online]. Available: https://github.com/USEPA/WNTR

[7] R. Puust, Z. Kapelan, D. A. Savić, and T. Koppel, "A review of methods for leakage management in water distribution systems," *Urban Water Journal*, vol. 7, no. 1, pp. 25–45, 2010.

[8] S. R. Mounce, J. Machell, and J. B. Boxall, "Burst detection in district metered areas using multivariate SPC," *Journal of Hydroinformatics*, vol. 12, no. 2, pp. 198–210, 2010.

[9] F. Nejjari, J. Meseguer, R. Sarrate, V. Puig, and G. Cembrano, "Leak localization in water distribution networks using a Bayesian approach," *Control Engineering Practice*, vol. 44, pp. 116–123, 2015.

[10] H. Mala-Jetmarova, N. Sultanova, and D. A. Savić, "Optimization of water distribution systems: a systematic review," *Water*, vol. 9, no. 11, p. 806, 2017.

[11] T. M. Walski *et al.*, *Advanced Water Distribution Modeling and Management*, 2nd ed. Exton, PA: Bentley Institute Press, 2016.

[12] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, *Guide to Industrial Control Systems (ICS) Security*, NIST SP 800-82 Rev. 2, 2015.

[13] International Electrotechnical Commission, *IEC 62443: Industrial Communication Networks—Network and System Security*, 2013–2018.

[14] Supabase, "Auth, Database, and Realtime Documentation," 2025. [Online]. Available: https://supabase.com/docs

[15] PostgreSQL Global Development Group, "PostgreSQL Documentation," 2024. [Online]. Available: https://www.postgresql.org/docs/