Politecnico di Torino

Master's degree in Digital Skills for Sustainable Societal Transitions

A.y. 2025/2026

Graduation Session February 2026

Design and Implementation of a Multi-Scale Decision Support Interface for Scenario-Based Energy Planning Web Platform

Supervisor:
Prof. G.Mutani

Candidate:
Mehrnaz Khosravi

Abstract

Territorial energy planning requires public administrations to interpret complex energy data, indicators, and scenarios across multiple spatial and temporal scales. While European frameworks such as National Energy and Climate Plans (NECPs) and Sustainable Energy and Climate Action Plans (SECAPs) define standardized indicators and reporting requirements, existing planning practices often rely on static documents and fragmented analytical tools, limiting interpretability, comparability, and transparency in decision-making.

This thesis presents the design and implementation of a multi-scale, scenario-based decision-support interface for territorial energy planning. The system is developed within a Design Science Research (DSR) framework and is explicitly aligned with SECAP methodologies and human–computer interaction (HCI) principles. The proposed dashboard integrates spatial data, energy indicators, and scenario logic into an interactive web-based interface that supports exploratory analysis, cross-administrative-level comparisons, and structured reasoning about alternative planning assumptions.

The research focuses on frontend design as a critical mediator between analytical systems and public-sector decision-makers. Key contributions include the formalization of multi-scale aggregation logic, transparent definitions of scenarios and indicators, and the integration of map- and chart-based visual analytics to support sense-making rather than automated decision-making. The system enables planners to examine energy consumption, renewable production, self-sufficiency, self-consumption, surplus, and uncovered demand consistently across municipalities, provinces, and regions.

The artifact is assessed using a comprehensive approach that combines analytical validation of indicators, inspection-based HCI evaluation, and checks for scenario consistency and sensitivity. Results show that the system effectively supports key territorial planning functions, including spatial prioritization, scenario comparison, and reporting, while ensuring analytical accuracy and clarity.

Overall, this work demonstrates that adequate decision support for regional energy planning comes from carefully designed analytical environments that emphasize transparency, comparability, and human-centered interaction. The thesis provides design valuable knowledge for future public-sector decision-support systems working in complex, multi-scale policy settings.

Content Table

Chapter 1 – Introduction

1.1 Background and Context

Energy systems across Europe are undergoing a major transformation driven by the need to combat climate change, guarantee energy security, and expand the share of renewable energy sources. This shift is reflected in coordinated policy frameworks that call for action and accountability at multiple levels of government, from national to regional and local authorities. In this context, territorial energy planning has become a vital component of sustainable development strategies, particularly within urban and regional governance.

Territorial energy planning examines energy consumption patterns, local production capacities, and future development potential within designated administrative areas, such as regions, provinces, and municipalities. These processes are inherently complex because they require integrating diverse datasets across spatial, temporal, and thematic scales. Decision-makers must consider technical, environmental, economic, and social factors simultaneously, while operating within regulatory constraints and aligning with long-term policy goals.

Although the availability of energy-related data has increased substantially in recent years, encompassing detailed consumption records, renewable energy production statistics, and scenario-based forecasts, public administrations often find it challenging to use this information effectively. Data are typically distributed across multiple agencies, stored in various formats, or communicated via static reports that constrain analytical flexibility. As a result, opportunities for exploration, comparison, and deeper understanding are often limited.

In this context, digital decision-support tools are becoming more important for enabling evidence-based energy planning. However, their success depends not only on data access and computational power but also on how information is displayed and accessed. In particular, the user interface design plays a key role in helping users navigate complex information, compare scenarios, and interpret key indicators clearly and meaningfully. The interface thus acts as a crucial link between analytical tools and human decision-making.

1.2 Motivation and Problem Statement

Various analytical models and simulation tools have been developed to analyze energy systems. Although many of these tools are advanced and generate detailed results, they are primarily designed for expert users and often require specialized knowledge to operate and interpret. In contrast, territorial energy planning typically involves a broader group of stakeholders, including municipal officials, planners, and policymakers, who may not possess advanced technical skills but are still responsible for making decisions with long-term impacts on their regions.

A common challenge in current planning methods is the gap between data availability and its practical application. Energy information is typically presented in static tables, summarized indicators, or lengthy reports that lack interactive capabilities or systematic scenario comparison. This situation hampers decision-makers' ability to identify spatial differences, track temporal changes, or evaluate trade-offs among development options. Specifically, predefined energy scenarios, commonly used in European planning, are often difficult to compare clearly and efficiently.

The multi-scale nature of territorial energy planning adds further complexity. Decisions made at the regional level can have different implications when viewed at the provincial or municipal scales, and highly aggregated indicators can mask local patterns that are crucial for effective policymaking. Existing tools often lack integrated mechanisms for navigating different spatial levels, forcing users to rely on disconnected analyses or multiple data sources. This fragmentation reduces transparency and increases the mental effort needed to interpret results.

Furthermore, key performance indicators used in energy planning, such as energy self-sufficiency, self-consumption, renewable energy penetration, and related economic or environmental metrics, are not always presented in a manner that facilitates interpretation and comparison. When indicators are shown without sufficient visual or contextual framing, their meaning can become unclear, limiting their usefulness for decision-making.

The main issue addressed in this thesis is the lack of user-centered, scenario-based interfaces that enable transparent, consistent exploration of territorial energy data across multiple scales. Solving this problem requires careful consideration not only of technical integration but also of interface design choices that align with planning needs and decision-making practices.

1.3 Research Objectives

This thesis focuses on developing a decision-support interface to facilitate the exploration and comparison of territorial energy data within scenario-based planning processes. The work aims to improve the usability and analytical accessibility of complex energy information for public administrations and planning stakeholders at the regional, provincial, and municipal levels.

To achieve this objective, the thesis focuses on understanding the specific requirements of territorial energy planning, particularly with respect to multi-scale analysis and the assessment of predefined energy scenarios. The interface is designed to support coherent navigation across territorial levels, allowing users to move between aggregated and localized views while maintaining contextual continuity.

An additional objective is to enable a structured comparison of alternative energy development scenarios through interactive controls and coordinated visualizations. By enabling scenario-based exploration, the interface is intended to support decision-makers in examining the implications of different planning assumptions and policy pathways, rather than relying on static indicators presented in isolation.

The thesis further aims to integrate and visualize key performance indicators commonly used in energy planning, including energy self-sufficiency, self-consumption, and selected environmental and economic metrics. Particular attention is given to ensuring that these indicators are presented transparently and clearly, reducing the risk of misinterpretation and supporting informed assessment.

Finally, the work applies principles of user-centered, explainable visualization to interface design, ensuring that analytical results remain accessible to non-technical users while remaining relevant for planning purposes. Through the design, implementation, and qualitative evaluation of the interface, the thesis demonstrates how frontend design decisions can significantly influence the effectiveness of decision-support tools in territorial energy planning.

1.4 Scope and Limitations

The scope of this thesis is deliberately limited to the design and implementation of the frontend interface for a decision-support tool in territorial energy planning. The work assumes an underlying backend infrastructure that provides access to energy datasets, scenario definitions, and computed indicators via appropriate application programming interfaces (APIs).

Consequently, the thesis does not address the development of energy simulation models, optimization methods, or data ingestion and processing pipelines. Similarly, the accuracy and validity of the underlying datasets and scenario assumptions are taken as given and not assessed within the scope of this research. The primary contribution of the thesis lies in the structure, presentation, and accessibility of the data and analytical results to end users.

The evaluation of the proposed interface is qualitative and focuses on its ability to support planning-relevant tasks, enhance interpretability, and enable scenario comparison. Comprehensive usability testing with large user groups and quantitative performance evaluation are beyond the scope of this work, but are identified as potential areas for future investigation.

Despite these limitations, the defined scope is consistent with the objectives of Design Science Research. It reflects practical conditions in territorial energy planning, where the adoption and effectiveness of analytical tools are often strongly influenced by the quality of interfaces.

1.5 Research Methodology Overview

This research is conducted within the Design Science Research (DSR) framework, which provides a structured approach to addressing practical problems through the design and evaluation of artifacts. DSR is particularly well-suited to research that bridges theoretical knowledge and practical application by producing solutions that are both methodologically sound and operationally relevant.

Within this framework, the decision support interface developed in this thesis serves as the primary research artifact. The research process follows the main stages of DSR: problem identification, objective definition, artifact design and development, demonstration, and evaluation. Each stage is informed by the specific characteristics of territorial energy planning and by principles of user-centered interface design.

Rather than testing hypotheses or deriving general laws, the research focuses on generating design knowledge through the systematic development of an interface that addresses a clearly defined problem. The artifact is evaluated by applying it to representative planning scenarios and assessing how well it meets the defined objectives.

1.6 Structure of the Thesis

The rest of this thesis is organized as follows. Chapter 2 examines the policy and planning context, focusing on European energy and climate frameworks and their implications for territorial energy planning. Chapter 3 examines related work on decision-support systems, scenario-based energy analysis tools, and visualization platforms, highlighting their interface limitations. Chapter 4 details the research methodology, emphasizing the use of the Design Science Research paradigm. Chapter 5 provides an overview of the system and data context, clarifying the role of the interface within a larger analytical infrastructure. Chapter 6 describes the design of the decision support interface, while Chapter 7 covers its frontend implementation. Chapter 8 showcases and evaluates the interface through representative use cases. Chapter 9 discusses the research results, contributions, and limitations. Finally, Chapter 10 concludes the thesis and suggests directions for future research and development.

Chapter 2 – Policy and Planning Context

2.1 European Energy and Climate Frameworks

European energy and climate policy provides the institutional and regulatory context for territorial energy planning. The current framework is primarily defined by the European Green Deal and the EU Climate Law, which set binding targets to reduce greenhouse gas emissions, increase the share of renewable energy, and improve energy efficiency (European Commission, 2019; European Union, 2021). These objectives require coordinated action across governance levels and are monitored through standardized reporting mechanisms.

Energy and climate planning at the European level relies heavily on quantitative data, scenario-based assessments, and performance indicators. Member States are required to collect and report energy data using harmonized methodologies, primarily coordinated by Eurostat and the European Environment Agency (Eurostat, 2023; EEA, 2023). These datasets serve as the empirical basis for policy monitoring and evaluation and are subsequently reused at national, regional, and local levels.

Within this framework, territorial energy planning represents the operational layer that translates European objectives into concrete actions. Regional and local authorities are therefore required to interpret policy targets using spatially explicit data, indicators, and scenarios, underscoring the importance of analytical tools that support interpretation across territorial scales.

2.1.1 National Energy and Climate Plans (NECPs)

National Energy and Climate Plans (NECPs) constitute the primary instruments through which EU Member States define their medium- and long-term strategies for energy and climate action. NECPs are mandated by Regulation (EU) 2018/1999 on the Governance of the Energy Union and Climate Action, which specifies both the structure of the plans and the associated reporting requirements (European Union, 2018).

NECPs are based on a combination of historical energy statistics, projections, and scenario analyses. Data sources typically include national statistical institutes, transmission system operators, and European datasets coordinated by Eurostat and ENTSO-E for electricity-related information (Eurostat, 2023; ENTSO-E, 2023). Scenario analysis is used to evaluate alternative policy pathways, such as different levels of renewable energy deployment, electrification, or demand-reduction measures.

Although NECPs are developed at the national level, their implementation depends on actions undertaken at the regional and local levels. Energy consumption, production, and infrastructure are territorially distributed, meaning that national indicators often require disaggregation or reinterpretation to support subnational decision-making. This creates demand for interfaces that enable territorial actors to map national objectives and scenarios to their specific administrative and spatial contexts.

2.1.2 Sustainable Energy and Climate Action Plans (SECAPs)

At the municipal level, Sustainable Energy and Climate Action Plans (SECAPs) constitute a key planning instrument for the local energy transition. SECAPs are developed within the framework of the Covenant of Mayors for Climate and Energy and follow standardized methodological guidelines provided by the Joint Research Centre (JRC) of the European Commission (European Commission, JRC, 2018).

SECAPs require municipalities to compile a Baseline Emission Inventory (BEI), typically based on local energy consumption data disaggregated by sector and combined with national or European-level emission factors. Familiar data sources include local utility providers, national energy agencies, and European reference datasets published by the JRC and the European Environment Agency (EEA, 2023).

Monitoring and reporting under SECAPs rely on a defined set of indicators, including final energy consumption, renewable energy production, greenhouse gas emissions, and measures related to local energy autonomy, such as self-consumption and self-sufficiency. Although these indicators are reported using standardized templates, they are often presented in static tabular or graphical formats, which limits their analytical depth and comparative value. This limitation underscores the need for digital tools that support interactive exploration, multi-scale comparison, and scenario-based assessment at the local level.

2.2 Territorial Energy Planning Challenges

Territorial energy planning is challenged by the need to operate simultaneously across multiple governance levels, each associated with different responsibilities, data resolutions, and decision-making processes. In Italy, municipalities typically manage detailed but often incomplete local data, while provinces and regions rely on more aggregated datasets derived from national statistics (Eurostat, 2023).

A central difficulty lies in interpreting indicators across spatial scales. Indicators such as renewable energy share, energy self-sufficiency, or sectoral consumption patterns may be meaningful at the national or regional level but require careful contextualization when applied locally. Aggregated values can obscure local disparities, while highly granular data can be challenging to interpret without appropriate analytical support.

In addition, many indicators used in NECPs and SECAPs are composite, involving relationships among multiple variables and assumptions. Without adequate visual explanation and contextual metadata, such indicators risk being misunderstood or misused, reducing their effectiveness as decision-support tools. Practical constraints within public administrations, including limited technical capacity, time pressure, and reliance on external consultants for data processing and analysis, further amplify these challenges.

2.3 Role of Digital Tools in Energy Planning

Digital tools play an increasingly important role in addressing the analytical, organizational, and communicative demands of contemporary energy planning. European institutions explicitly encourage the use of digital platforms to support monitoring, reporting, and stakeholder engagement in energy and climate policies (European Commission, 2020).

Digital interfaces are particularly relevant because they serve as the point of interaction between complex datasets and human decision-makers. Well-designed interfaces allow users to navigate across spatial and temporal dimensions, compare indicators and scenarios, and interpret results derived from official data sources without requiring direct interaction with underlying models or databases.

For both NECPs and SECAPs, where transparency, accountability, and consistency across governance levels are central requirements, digital decision-support tools can improve coordination between institutions and facilitate more coherent planning practices. In particular, multi-scale digital dashboards can support evidence-based decision-making by linking territorial data, analytical logic, and visualization within a unified system.

3.2 Problem Definition

Territorial energy planning in Italy requires public administrations to operate simultaneously across multiple governance levels, each associated with distinct responsibilities, data resolutions, and decision-making processes. As described in Chapter 2, municipalities typically work with detailed but often incomplete local datasets, while provinces and regions rely on more aggregated information derived from national statistics. This structural fragmentation creates significant challenges for evidence-based decision-making.

A central problem lies in the **interpretation of energy indicators across spatial scales**. Indicators such as renewable energy share, energy self-sufficiency, and sectoral consumption patterns are commonly defined and reported at the national or regional level. However, they must be meaningfully applied at the municipal scale to support local planning actions. Aggregated indicators may conceal substantial territorial disparities, whereas highly granular data can be challenging to interpret without appropriate analysis and contextualization.

In addition, many indicators used in National Energy and Climate Plans (NECPs) and Sustainable Energy and Climate Action Plans (SECAPs) are **composite**, relying on relationships among multiple variables, assumptions, and reference values. When these indicators are presented through static tables or figures, their underlying structure and implications are not always transparent. This increases the risk of misinterpretation and reduces their effectiveness as decision-support instruments.

Organizational and practical constraints within public administrations further compound these analytical challenges. Limited technical capacity, time pressure, and reliance on manual data processing or external consultants often lead to fragmented analytical workflows and low reproducibility of results. As a consequence, answering fundamental planning questions—such as identifying priority territories, comparing performance across administrative levels, or assessing the implications of alternative scenarios—becomes time-consuming and error-prone.

The problem addressed in this thesis can therefore be summarized as follows:

Public administrations require consistent, transparent, and scalable analytical support to interpret energy indicators across multiple territorial levels, but existing data infrastructures and tools do not adequately support this need.

This problem definition directly motivates the design of a digital decision-support system that integrates heterogeneous energy datasets, applies consistent aggregation logic across spatial and temporal scales, and presents results through interactive, interpretable visual interfaces. The design requirements derived from this problem are discussed in the following sections.

2.4 Summary

This chapter outlines the policy and planning context for territorial energy planning, highlighting the role of European frameworks such as NECPs and SECAPs, as well as the official data sources on which they rely. Energy planning processes are increasingly data-driven and indicator-based, drawing on harmonized datasets from institutions such as Eurostat, the European Environment Agency, and the Joint Research Centre.

Challenges associated with multi-scale governance, indicator complexity, and limited administrative capacity reveal the shortcomings of traditional planning approaches grounded in static documentation. These considerations directly motivate the need for digital decision-support interfaces that make policy-relevant data accessible, interpretable, and comparable across territorial scales. The requirements identified in this chapter inform the interface design choices discussed in the following chapters.

3. Methodology

3.1 Research paradigm and methodological positioning

This research uses a design science research (DSR) approach, mainly to create and assess a software artifact that tackles a real-world problem in municipal energy analysis. Design science is particularly appropriate when the goal is to develop artifacts that provide practical solutions while advancing knowledge in information systems (Hevner et al., 2004). The study adopts a pragmatic approach, focusing on the artifact's usefulness and real-world applicability rather than testing hypotheses or establishing causality. In line with design-oriented research, knowledge is generated through iterative cycles of building and testing the artifact in a real setting (March & Smith, 1995). Methodologically, the research follows the Design Science Research Methodology (DSRM) proposed by Peffers et al. (2007), which guides the process through problem identification, goal Setting, artifact design and development, demonstration, and evaluation. The evaluation aims for analytical generalization, meaning insights from the design inform similar decision-support systems rather than relying on statistical generalization. Overall, this study is part of information systems research, emphasizing data-driven decision support and interactive visualization for local energy planning.

3.2 Framework of Design Science Research

This research adopts the Design Science Research (DSR) paradigm, which considers the creation and evaluation of artifacts as rigorous and valid scientific methods in the field of information systems. While natural or behavioral research primarily aims to explain or predict phenomena, DSR focuses on designing solutions to specific problems and generating knowledge through the design process itself (March & Smith, 1995; Hevner et al., 2004). A key premise of DSR is that complex socio-technical challenges, such as territorial energy planning, cannot be addressed through observation alone. Instead, they require developing artifacts that incorporate domain expertise, support decision-making, and can be systematically tested in real-world settings. In this study, the artifact is a multi-scale, scenario-based digital decision-support system that integrates spatial data, energy indicators, and interactive visual analytics to assist public administrations in planning and assessment tasks.

Building on the foundational framework by Hevner et al. (2004), this research aligns with core DSR principles: (i) creating a purposeful artifact, (ii) addressing a real-world problem, (iii) grounding in existing knowledge, (iv) thorough evaluation, and (v) clear communication of insights. These principles balance relevance—tackling a genuine planning issue faced by municipalities and regions—and rigor, with design choices rooted in established theories and methods. The difference between design science and natural science, as explained by March and Smith (1995), provides an additional conceptual foundation. While natural science describes and explains current systems, design science aims to develop artifacts such as constructs, models, methods, and instantiations that improve organizational or decision-making capabilities. This thesis presents the main research contributions through a mix of artifacts (the system), models (scenario representations and indicator aggregation logic), and design principles for future decision-support systems in similar contexts. The study employs the Design Science Research Methodology (DSRM) outlined by Peffers et al. (2007), which guides the research process in stages: defining the problem and motivation, setting solution goals, designing and developing the solution, demonstrating its application, evaluating its effectiveness, and sharing the results. Although these stages are iterative rather than linear, insights from later phases can influence earlier decisions. Consequently, the thesis follows this methodology: early chapters describe the decision context and design goals, followed by system development and implementation, ending with evaluation and reflections.

Evaluation is a core aspect of the DSR paradigm, viewed as an integral part of the design process rather than just a final validation step. According to the three-cycle model of DSR (Hevner, 2007), this research addresses the relevance cycle by anchoring requirements in actual planning practices, the rigor cycle by utilizing established literature in energy planning, HCI, and visual analytics, and the design cycle through iterative development and testing of the artifact. The evaluation approach combines analytical validation of indicators, interface inspections, and scenario consistency checks, aligning with evaluation-focused DSR frameworks (Venable et al., 2016). Furthermore, DSR is particularly apt for the public sector's policy-focused context in territorial energy planning. Instead of solely optimizing one quantitative measure, the system facilitates understanding, comparison, and exploration across various spatial scales and scenarios. The contribution of this research extends beyond the software artifact to include the design principles, structures, and interaction strategies that can be applied in other territorial decision-support settings.

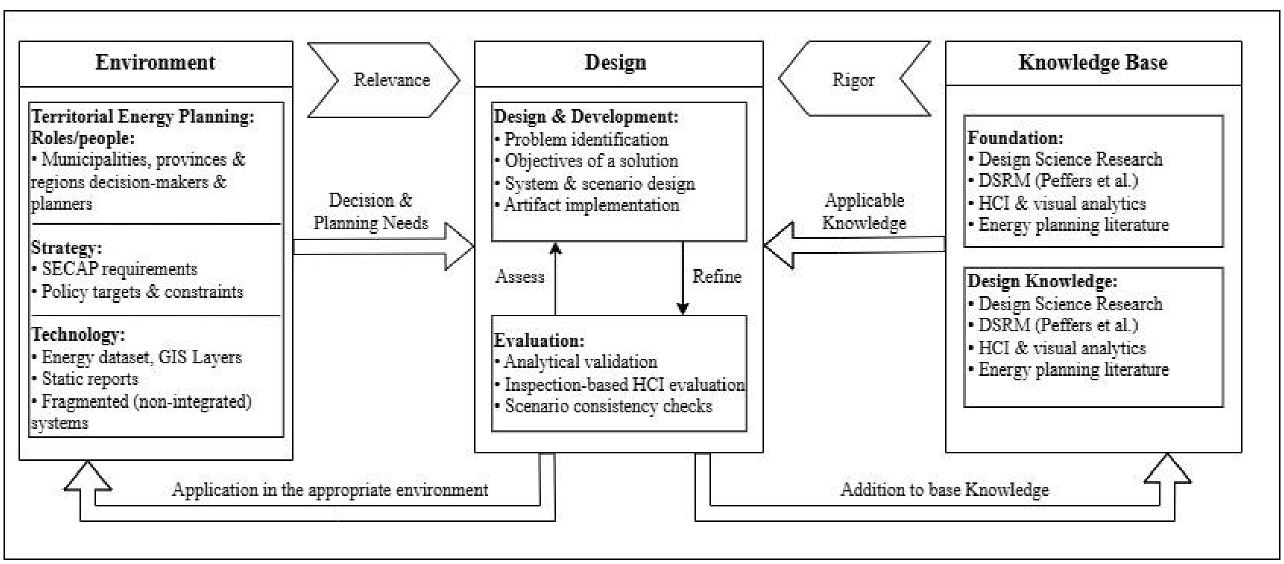

Figure 3.1. Methodological positioning of the research. This study adopts Design Science Research (DSR) as the The overarching paradigm (Hevner et al., 2004) is implemented through the Design Science Research Methodology (DSRM) (Peffers et al., 2007). The design cycle facilitates an iterative process. artifact development and evaluation, while the relevance and rigor cycles ensure grounding in planning practice, and established scientific Knowledge is represented through constructs, models, methods, and instantiations (March & Smith, 1995).

3.3 Problem Identification and Decision Context

This research investigates a decision-making context that differs from the policy and planning environment discussed in Chapter 2. Although European and national frameworks establish goals, indicators, and reporting standards, actual territorial energy planning relies on how public administrations interpret and apply these guidelines in real decision situations. The study focuses on public sector officials and planners at municipal, provincial, and regional levels. Their role is not to develop technical energy models or perform detailed calculations but to interpret territorial data, compare administrative units, and justify planning decisions within institutional, regulatory, and political constraints. Decision-making in this context is affected by limited time, incomplete information, and the need for simplified representations rather than comprehensive analysis—a common trait in decision theory (Simon, 1955). Here, decision-making is mainly exploratory and interpretative, not strictly deterministic or optimized. Planning involves identifying territorial differences, analyzing trends, prioritizing intervention areas, and evaluating the effects of different assumptions. These activities emphasize sense-making and comparative judgment, reflecting the nature of strategic decision-making in public organizations.

A key implication of this decision-making environment is that complex analytical information must be presented through representations that humans can interpret easily. Decision-makers do not work directly with raw data or models; instead, their understanding is influenced by summaries, indicators, and visual abstractions. Human-computer interaction research shows that effective decision support relies on interfaces that minimize cognitive effort, make relationships clear, and allow users to explore data gradually without needing technical expertise (Norman, 2013; Shneiderman et al., 2016). From a Design Science Research perspective, this highlights a clear and well-defined problem: the gap between the complexity of territorial energy planning and the cognitive and practical abilities of its users. Closing this gap involves creating decision-support tools that emphasize interpretability, comparability, and usability, helping public-sector actors engage effectively with energy data in their routine planning and decision-making (Hevner et al., 2004).

3.4 SECAP as Domain Justificatory Knowledge

Within the framework of Design Science Research, domain justificatory knowledge encompasses the established concepts and structures that underpin artifact design within a specific field. In this study, Sustainable Energy and Climate Action Plans (SECAPs) serve as the primary source of such knowledge, outlining the standard approach municipalities use to evaluate, plan, and track energy and climate initiatives.

SECAP guidelines mandate a structured reporting system focused on the quantitative analysis of energy use, local renewable energy generation, and resulting greenhouse gas emissions. These data are initially recorded in the Baseline Emission Inventory and then reused to set targets, assess scenarios, and support ongoing monitoring. Consequently, SECAPs establish a uniform set of indicators and relationships that guide local authorities in interpreting energy performance.

From an analytical perspective, SECAP reporting can be simplified into a few key data categories: territorial energy consumption, renewable energy output by technology, performance metrics such as renewable share or energy self-sufficiency, and comparisons over time between baseline and monitoring years. These core elements form the informational basis for SECAP-driven decision-making and are standardized across municipalities.

The decision-support system developed in this project aligns directly with the SECAP core logic. Instead of focusing on narrative or administrative details, it emphasizes quantitative data that facilitate comparisons, interpretation, and scenario analysis across territorial levels. It operationalizes SECAP concepts within a digital platform, making energy performance indicators more accessible and easier to interpret, while fully adhering to established SECAP methodologies.

SECAP data group	Core content	Analytical characteristics	Dashboard support
Energy consumption (Group B)	Final energy consumption by sector and carrier	Aggregated by territory; comparable across administrative scales; explorable across time.	Multi-scale aggregation (municipality, province, region); temporal exploration and comparison
Renewable energy production (Group C)	Local renewable production by technology (e.g., solar, wind, hydro, biomass)	Technology-specific; spatially distributed; comparable to consumption values	Spatial visualization and cross-comparison with consumption indicators
Derived indicators (Group D)	Renewable energy share; energy balance; self-sufficiency / autonomy; emission-related indicators (derived)	Computed from consumption and production data; interpretative indicators	On-the-fly indicator computation and comparative assessment
Temporal comparison (Group E)	Baseline and monitoring values; scenario assumptions	Longitudinal comparison; trend detection; scenario-based reasoning	Baseline vs monitoring comparison; trend and scenario exploration

Table 3.1 — SECAP data categories and available analytical tools

The decision-support system created in this research focuses specifically on key SECAP data groups. Rather than emphasizing narrative, governance, or financial details, it highlights the quantitative elements critical for interpretation, comparison, and scenario analysis at different territorial levels. This method helps translate SECAP concepts into a digital format while remaining fully aligned with established SECAP methodologies. Such alignment ensures the tool is more than a simple analytical resource; it is deeply rooted in current planning practices and reporting standards, enhancing the robustness and relevance of the design choices discussed in the following sections.

3.5 Human–Computer Interaction as Design Justificatory Knowledge

This research applies Human–Computer Interaction (HCI) to establish foundational design principles for a decision-support tool that manages complex territorial energy systems. The information is tailored to the target users. Since the decision context in Section 3.3.3 is exploratory and interpretive, the study applies established HCI principles to ensure that the system supports understanding, comparison, and judgment rather than technical analysis.

A core principle is to minimize cognitive load. HCI research emphasizes that decision-support systems should reduce unnecessary mental effort, allowing users to focus on interpretation instead of navigating the interface (Norman, 2013). This is done by organizing information progressively, distinguishing core indicators from derived measures, and avoiding overwhelming users with excessive analytical details at once. Complexity is addressed through aggregation, abstraction, and step-by-step exploration, rather than exhaustive displays.

A second principle focuses on supporting comparison and pattern recognition. Interactive systems meant for analysis should help users identify similarities, differences, and trends across multiple dimensions, such as space and time (Shneiderman et al., 2016). This research emphasizes comparison as a primary interaction goal: energy consumption, renewable production, and derived indicators are consistently organized for comparison across territorial scales, time periods, and scenarios.

A third principle emphasizes user control and direct interaction. HCI research highlights the importance of allowing users to explore information through direct manipulation, rather than passive viewing (Shneiderman et al., 2016). In this context, users can select territories, change time perspectives, and switch views without predefined workflows or technical commands.

Finally, the principle of interpretability and transparency guides the design. For public-sector decision-support systems, users must understand how indicators are derived and how representations relate to underlying concepts. This research adheres to this principle by grounding all analyses in established SECAP concepts and maintaining a clear link between raw measures (e.g., consumption and production) and derived indicators (e.g., renewable share or energy balance).

Together, these HCI principles form the basis for the design decisions in this study. They guarantee that the artifact facilitates decision-making as a process of sense-making and comparison, in line with the cognitive traits and practical limitations of territorial energy planning. These principles directly inform the design and development process detailed in the following section.

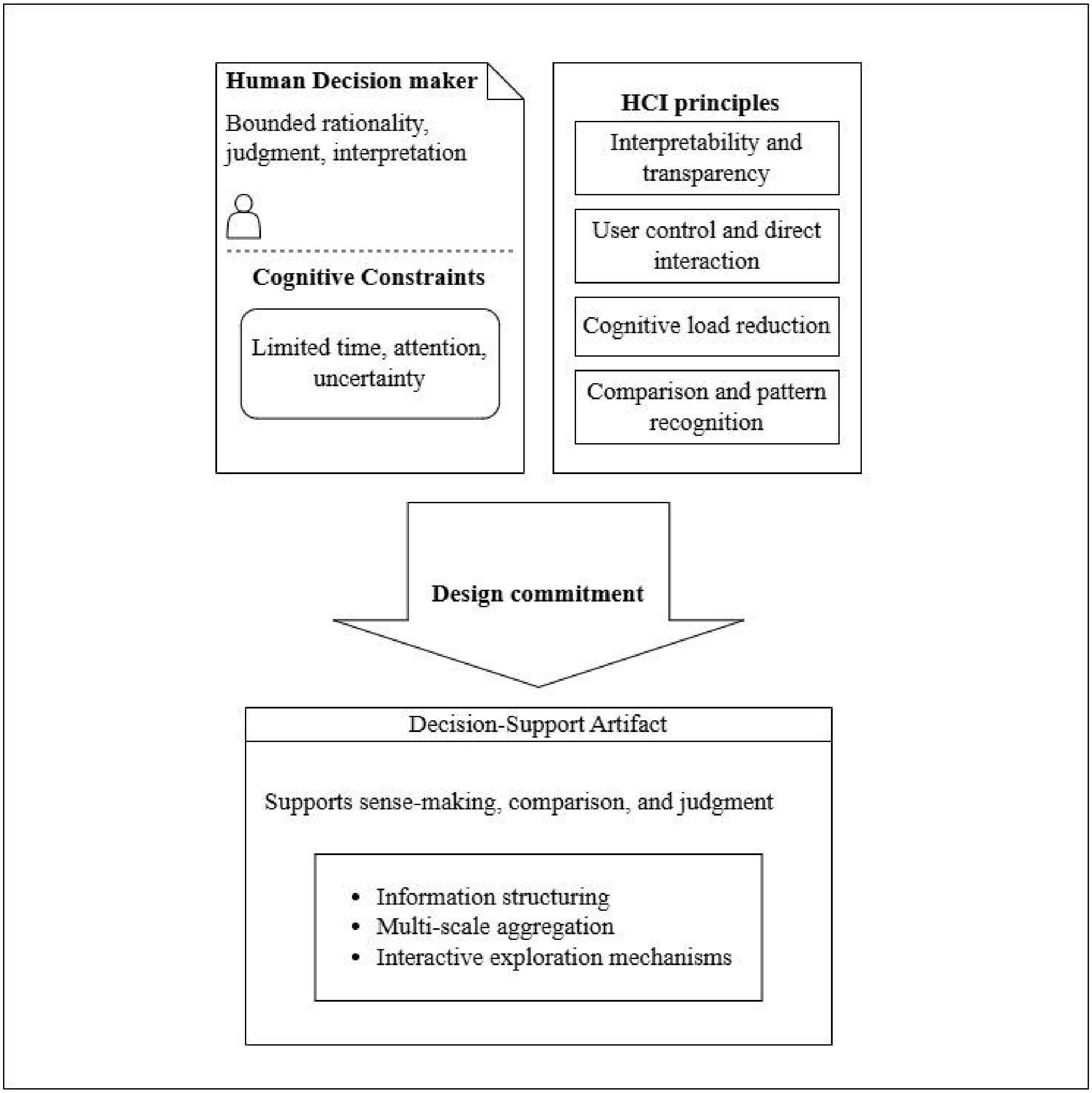

Figure 3.2 — Human-centered decision-support logic guides the design of the proposed artifact. The diagram shows how the cognitive traits of human decision-makers influence HCI principles, which in turn shape specific design choices embodied in the decision-support artifact.

Figure 3.2 depicts decision support as being human-centered, drawing on decision theory and HCI research. Public-sector planning decisions are hindered by bounded rationality, causing decision-makers to depend on simplified information and judgment instead of detailed analysis (Simon, 1955). This inspires HCI principles that lessen mental effort and promote exploratory interaction, allowing users to navigate complex information gradually (Norman, 2013; Shneiderman et al., 2016). How information is displayed influences reasoning and comprehension, as demonstrated by models of human information processing and visualization (Card, Moran, & Newell, 1983; Ware, 2013). The focus of this research is on organizing information, facilitating comparisons, and enabling direct interaction to handle complexity. In line with Design Science Research, the decision-support tool is intended not to automate decisions but to support human decision-making by improving sense-making and judgment (Hevner et al., 2004).

3.7 Scenario-Based Decision-Support Logic

Scenario-based decision-making is a planning method for decisions under uncertainty and trade-offs. It doesn't seek a single "optimal" answer but compares plausible alternatives and their outcomes. Scenarios are structured counterfactuals: each has assumptions and parameters applied to the same energy baseline, enabling transparent comparisons across space, time, and stakeholders (van der Heijden, 2005; Trutnevyte et al., 2019). This aligns with SECAP practice, where decision-makers analyze baseline and monitoring cycles and assess progress using indicators rather than relying on predictive optimization.

3.7.1 Scenario definition and scope

This thesis defines a scenario as a coherent set of assumptions and parameters that influence territorial energy flows—both demand and supply—and/or organizational structures like energy communities. These scenarios generate comparable indicators across different spatial scales and reporting periods. They are designed primarily for decision support rather than forecasting. Their main goal is to improve the interpretability, traceability, and comparability of planning options, consistent with how scenarios are used in energy and climate policy contexts (Trutnevyte et al., 2019; Pfenninger, et al., 2014).

3.7.2 Scenario dimensions

Each scenario is outlined through four key methodological dimensions. This separation is crucial for ensuring traceability from assumptions to outcomes.

(A) Demand-side configuration (consumption):

- Sectoral electricity demand (e.g., residential, tertiary, industrial/public services)

- Optional demand variation assumptions (e.g., efficiency improvements, electrification effects), where supported by the dataset and case study scope

(B) Supply-side configuration (renewable production):

- Current production (observed) by renewable technology

- Projected future production (scenario) based on spatial potential assessment (see Section 3.7.4)

(C) Organizational configuration (energy communities and sharing):

- Rules/assumptions regarding the allocation of locally produced electricity across consumer sectors

- Scenario types indicating various levels of collective self-consumption or internal balancing

(D) Temporal configuration (reporting horizon):

- Annual reporting for year-by-year comparison

- Medium-term comparison aligned with SECAP (e.g., 4-year monitoring intervals) to facilitate reporting and progress evaluation

This structure reflects the literature's emphasis that scenarios must clearly specify what changes are assumed, what remains fixed as a baseline, and what is measured as indicators; otherwise, comparisons become unclear (van der Heijden, 2005; Trutnevyte et al., 2019).

3.7.3 Indicator set per scenario

To maintain consistency with SECAP logic while offering decision-support value, each scenario generates a fixed indicator panel grouped systematically. This approach is methodologically crucial: it prevents "indicator drift" across different scenarios and ensures they are comparable.

Group 1 — Energy accounting (SECAP-consistent core)

These indicators display the main energy status of a region for a specific year and scale:

- Electricity demand (by sector and total)

- Renewable electricity generation (by technology and total)

- Net balance (production minus demand)

Group 2 — Autonomy and local use

These indicators assess "the extent to which local demand is satisfied locally":

- Self-sufficiency (local RES used divided by total demand)

- Self-consumption (local RES used divided by total local RES production)

Group 3 — Surplus and uncovered demand

These indicators clarify trade-offs and are essential for interpreting plans:

- Surplus (local RES production that is not used locally, but exported or excess)

- Uncovered demand (demand met externally, indicating imports or a deficit)

Group 4 — Environmental indicators (derived, SECAP-compatible)

Where emission factors and accounting conventions are available and applied consistently:

Emission-related indicators based on energy flows (reported as derived measures rather than raw inventory figures)

Group 5 — Economic indicators (decision-support level)

Economic representation is included only at a level consistent with methodology and data availability:

Indicative cost/benefit or proxy measures (e.g., avoided external energy purchase proxies, or cost ranges if scenario datasets support them) are used to support comparison and discussion, not to replace detailed techno-economic appraisal (Pfenninger et al., 2014).

This grouping reflects best practices in energy planning: multidimensional indicator sets improve the robustness and transparency of scenario comparisons, especially when decisions are socio-technical and value-laden (Trutnevyte et al., 2019).

3.7.3 Scenario indicator set: formal definitions and interpretation

To ensure methodological consistency and comparability across different scenarios, territories, and reporting periods, each scenario in the decision-support system produces a fixed set of indicators. These indicators are based on a small number of core energy variables and are calculated uniformly across spatial and temporal scales. Let technnology.

Base energy variables:

Total electricity demand in the territory $i$during year $t$is defined as the sum of sectoral demands:

$$D_{i,t} = \sum_{s} D_{i,t,s}$$

where $D_{i,t,s}$represents the electricity demand of the sector s. This quantity expresses the total energy requirement that must be satisfied either locally or through external supply.

Total local renewable electricity production is defined as:

$$P_{i,t} = \sum_{k} P_{i,t,k}$$

where $P_{i,t,k}$denotes electricity produced locally from renewable technology k. This represents the maximum renewable supply potentially available to the territory.

A key intermediate variable is the amount of locally used renewable electricity, denoted. $U_{i,t}$, which represents the portion of local renewable production that is actually consumed within the same territory. When energy-community scenarios are considered, this value can be further disaggregated by sector as $U_{i,t,s}$, with the constraint $0 \leq U_{i,t,s} \leq D_{i,t,s}$.

Energy accounting and balance:

The net energy balance of a territory is defined as the difference between renewable production and total demand:

$$B_{i,t} = P_{i,t} - D_{i,t}$$

A positive value of indicates that the territory is, on the whole, a net producer of renewable electricity, while a negative value shows reliance on external supply. Although this provides useful information, it does not reveal how much renewable energy is actually used locally, which is why the following indicators are important.tes the following indicators.

Self-sufficiency and self-consumption:

The self-sufficiency ratio measures the share of total demand that is covered by locally produced and locally used renewable electricity:

$$SS_{i,t} = \frac{U_{i,t}}{D_{i,t}}$$

This indicator measures the level of local energy independence. A value of ￼ indicates complete local coverage of demand by renewable sources, while lower values show increasing dependence on external sources supplydicate increasing reliance on external supply.

The self-consumption ratio measures how efficiently local renewable production is utilized within the territory:

$$SC_{i,t} = \frac{U_{i,t}}{P_{i,t}}$$

A high value ￼ suggests that most locally produced renewable electricity is used locally, while a low value indicates surplus generation that is exported or curtailed. Self-sufficiency and self-consumption offer complementary insights: the former emphasizes demand coverage, and the latter emphasizes how well production is utilized.

Surplus and uncovered demand:

To make trade-offs explicit, two absolute indicators are derived from the same base variables.

The surplus renewable energy is defined as:

$$S_{i,t} = \max\,(0,\, P_{i,t} - U_{i,t})$$

This refers to renewable electricity generated nearby but not consumed locally, indicating exports or possible curtailment.

The uncovered demand is defined as:

$$UD_{i,t} = \max\,(0,\, D_{i,t} - U_{i,t})$$

This represents the portion of demand that must be met by external energy sources, indicating residual dependency in the broader energy system.

These definitions ensure the following mass-balance identities hold for every scenario:

$$P_{i,t} = U_{i,t} + S_{i,t}, D_{i,t} = U_{i,t} + UD_{i,t}$$

These identities are enforced as methodological consistency checks.

For comparative purposes, surplus and uncovered demand can also be normalized as ratios:

$$SR_{i,t} = \frac{S_{i,t}}{P_{i,t}}, UR_{i,t} = \frac{UD_{i,t}}{D_{i,t}}$$

These normalized indicators allow comparison across territories of different sizes.

Sector-level indicators for energy communities:

When scenarios feature energy-community setups, sector-level indicators are calculated to clearly show distributional effects.

Sectoral self-sufficiency is defined as:

$$SS_{i,t,s} = \frac{U_{i,t,s}}{D_{i,t,s}}$$

This indicator shows the share of the sector's electricity demand met by locally allocated renewable energy, highlighting which sectors benefit most from collective arrangements.

The allocation share of renewable energy for the sector $s$is defined as:

$$AS_{i,t,s} = \frac{U_{i,t,s}}{U_{i,t}}$$

This expresses how locally used renewable energy is distributed among sectors within an energy community.

Multi-scale aggregation rule:

To preserve comparability across spatial scales, indicators for aggregated territories (e.g., provinces or regions) are computed by first summing the base variables and then deriving ratios. For an aggregate territory A:

$$D_{A,t} = \sum_{i \in A} D_{i,t}\, , P_{A,t} = \sum_{i \in A} P_{i,t}\, , U_{A,t} = \sum_{i \in A} U_{i,t}$$

3.7.4 Energy communities as an explicit scenario mechanism

A significant advancement in this research is the ability to represent energy community configurations across scenario variants. Methodologically, an energy community scenario outlines how locally generated renewable electricity is distributed across consumption sectors within a territory (or a meaningful administrative subset), producing metrics for self-consumption, self-sufficiency, surplus, and unmet demand. This is regarded not as a minor feature but as a structured scenario dimension, since recent studies indicate that energy communities affect participation, governance, local balancing, and perceived benefits (Lowitzsch et al., 2020; Roberts et al., 2019). In the dashboard, energy communities are thus shown as organizational scenarios that can be contrasted with non-community baselines through the same indicator panel (Section 3.7.3), allowing decision-makers to evaluate whether collective setups enhance local autonomy or alter surplus and deficit patterns.

3.7.5 Future production scenarios from spatial opportunity assessment

Future renewable energy production is modeled as a scenario component based on geographic suitability and spatial opportunity methods. This approach links planning choices to spatial constraints and potentials by identifying where additional renewable deployment is feasible and how it influences territorial energy balances. It aligns with the energy systems literature, which highlights the importance of considering spatial constraints, resource availability, land, and infrastructure in credible scenario analysis (Pfenninger et al., 2014), as well as with planning research emphasizing the role of spatial reasoning in the energy transition (Delmastro et al., 2020). In the dashboard framework, spatial opportunity layers help define scenarios (supply-side configuration), while resulting production levels are integrated into the same indicator system, ensuring traceability from geographic assumptions to decision-making metrics.

3.7.6 Temporal logic and SECAP-aligned comparison

To facilitate both operational monitoring and formal reporting, scenarios are assessed at: an annual resolution for analyzing year-to-year trends, and SECAP-aligned reporting intervals (such as four-year monitoring cycles), which provide a structured basis for baseline–monitoring comparisons within administrative planning processes. This dual approach in time framing allows the system to support immediate understanding and medium-term accountability, while avoiding the misleading precision often associated with long-term forecasts in uncertain contexts (Trutnevyte et al., 2019).

3.7.7 What does the scenario logic enable?

The scenario-based decision-support system allows decision-makers to consistently compare territories across different scales and years, analyze trade-offs among autonomy, surplus, and residual dependency, assess the impacts of energy community setups across sectors, and spot opportunities for additional clean energy generation grounded in spatial data. It also offers SECAP-aligned, traceable indicator summaries suitable for reporting cycles. Consistent with decision-support theory and Design Science Research principles, the system aims to enhance human judgment by organizing and revealing the consequences of different assumptions, rather than dictating choices or ranking scenarios as "best" (Hevner et al., 2004).

All ratio-based indicators (e.g., $SS_{A,t}$, $SC_{A,t}$) are then computed from these aggregated totals. This rule avoids scale-induced distortions and ensures methodological consistency when moving between municipal, provincial, and regional analyses.

By establishing all indicators with clear equations and applying mass-balance and aggregation constraints, the scenario-based approach guarantees that comparisons across different scenarios showcase genuine differences in assumptions and setups, rather than being influenced by computational or scale artifacts. This structured formalization is crucial for transparent decision-making, especially in institutional settings like SECAP reporting, where indicators need to be understandable, justifiable, and reproducible.

3.8 Integrated Evaluation, Worked Decision Vignette, and Derived Design Principles

The proposed decision-support system is evaluated using a comprehensive methodological approach that includes (i) formative, artifact-centered assessment, (ii) a realistic decision vignette based on territorial planning tasks, and (iii) the development of generalizable design principles. This approach aligns with Design Science Research (DSR), which assesses artifacts based on their ability to support specific decision types rather than solely on performance metrics or predictive accuracy (Hevner et al., 2004). Since the artifact aims to assist exploratory and interpretive decision-making in territorial energy planning, the evaluation focuses on whether the system enables decision-makers to analyze scenarios, interpret trade-offs, and justify planning decisions in a transparent and coherent way.

3.8.1 Evaluation strategy and methods

This research uses a scenario-based evaluation with decision tasks. Instead of focusing on user deployment or quantitative accuracy, the artifact is judged by whether it effectively supports key planning tasks derived from SECAP workflows. This method is common in DSR and Human–Computer Interaction studies, especially when systems aim to improve human judgment and understanding rather than automate decisions (Hevner et al., 2004; Shneiderman et al., 2016). The evaluation criteria include decision coverage—for addressing major questions like scenario comparison, local energy autonomy, and detecting spatial and temporal mismatches between demand and supply; internal consistency—ensuring indicators are correct and stable across various spatial scales, time frames, and scenarios, including compliance with mass-balance and aggregation rules; comparability—allowing scenarios, territories, and sectors to be compared with invariant indicators and consistent logic; interpretability—the ease of understanding indicator changes and linking them to assumptions and scenario parameters; and non-prescriptiveness—revealing trade-offs without ranking scenarios or suggesting preferred outcomes. The methodology involves analytical walkthroughs where the dashboard is tested against realistic planning questions, examining indicator behavior across scales, supporting scenario comparison through interaction mechanisms, and exploring different assumptions without ambiguity. The outcome is not a numerical score but evidence that the artifact is suitable for its intended decision-support role.

Decision Task	Primary Stakeholder	Decision Output	Evaluation Method (DSR-aligned)
Identify high-energy consumption areas.	Municipal technical staff	Priority territories for intervention	Analytical validation of aggregation + use-case walkthrough
Compare municipalities within a province	Provincial planners	Benchmarking and coordination insights	Consistency check across scales + comparative task walkthrough
Assess sectoral energy demand.	Municipal energy managers	Sector-specific policy focus	Indicator correctness check + interpretability inspection
Monitor progress vs baseline (BEI)	Municipal policy-makers	Assessment of policy effectiveness	Scenario-free analytical comparison
Evaluate the renewable energy potential distribution.	Regional planners	Identification of priority RES sources	Spatial coherence validation
Assess local energy autonomy.	Municipal decision-makers	Understanding dependency on external supply	Indicator logic validation + heuristic UI review
Compare alternative planning scenarios.	Policy-makers (strategic)	Selection of preferred planning pathway	Scenario plausibility check + sensitivity reasoning
Identify seasonal mismatches	Technical planners	Timing of interventions and storage needs	Temporal aggregation correctness
Support reporting and communication	Policy-makers	Transparent reporting material	Output consistency and clarity inspection
Explain results across governance levels.	Regional coordinators	Shared understanding across levels	Cognitive walkthrough (HCI-based)

Table 3.1 This text summarizes how decision tasks, stakeholder groups, dashboard features, and evaluation methods are interconnected in this study. Instead of examining each interface component separately, the evaluation centers on typical decision-making tasks frequently faced in territorial energy planning by public administrations. This task-focused approach aligns with Design Science Research principles and enables the assessment of the artifact's usefulness in realistic planning scenarios.

3.8.2 Worked decision vignette

To illustrate how the artifact aids decision-making, a detailed decision vignette is included in the evaluation. This vignette presents a realistic territorial planning task instead of a hypothetical scenario. It depicts a territorial planner using the interactive dashboard developed in this study to explore electricity planning options. The planner starts by choosing a territorial scale—municipality, province, or region—on the React-based interactive map. This map allows simultaneous visualization of electricity consumption across sectors (primary, secondary, tertiary, residential) and renewable energy sources (biomass, wind, geothermal, solar, hydro), facilitating quick spatial comparisons of demand and supply. The planner then activates different scenarios, including ones with increased future renewable production based on spatial opportunity assessments. As scenarios update, the dashboard dynamically adjusts indicators like self-sufficiency, self-consumption, surplus, and unmet demand. By toggling energy-community configurations, the planner examines how locally produced electricity redistributes across sectors and its effects on sector autonomy and surplus. Temporal analysis is enabled by comparing yearly data and SECAP-aligned four-year periods, helping the planner determine if improvements reduce external dependency or just shift surplus. The system does not suggest the best scenario but makes risks and trade-offs transparent, supporting the planner in forming and justifying decisions. This vignette demonstrates that the artifact enhances sense-making, comparison, and justification — essential aspects of real-world territorial energy planning.

3.8.3 Design principles derived from the study

Following the evaluation and the decision vignette, a set of design principles has been developed. These principles reflect broad insights that go beyond the particular technological implementation in this study..

- DP1. Human-centered scenario exploration
  Decision-support systems for energy planning should emphasize exploratory interaction and comparative analysis rather than automated optimization or prescriptive solutions.
- DP2. Multi-scale consistency by design
  Indicators must be derived from invariant base variables and aggregated in a manner that ensures coherence across municipal, provincial, and regional levels.
- DP3. Integrated spatial visualization of demand and supply
  Spatial interfaces should concurrently display consumption sectors and production technologies to facilitate reasoning about mismatches, synergies, and territorial opportunities.
- DP4. Clear depiction of surplus and unmet demand
  Surplus production and residual dependencies need to be made visible, as they are crucial for understanding renewable deployment impacts and energy-community configurations.
- DP5. Sector-aware modeling of energy communities
  Scenario logic should enable the redistribution of local renewable energy across consumption sectors to highlight distributional effects and collective self-consumption opportunities.
- DP6. Alignment with institutional reporting cycles
  Analytical views should support both annual monitoring and multi-year reporting aligned with SECAP to promote institutional relevance and accountability.

These principles show how the proposed artifact integrates domain knowledge (SECAP), human-centered design principles (HCI), and scenario-based reasoning into a single decision-support system. The evaluation verifies that the artifact successfully aids the targeted decision-making process, while also recognizing that long-term institutional adoption and empirical user studies are important areas for future research.

3.9 Threats to Validity and Limitations

As a Design Science Research study, this work faces multiple validity considerations due to the nature of decision-support artifacts, scenario-based reasoning, and human-centered analytical systems. Explicitly addressing these threats is crucial to accurately position the contribution and clarify the scope within which the results and design principles are valid.

3.9.1 Construct validity

Construct validity refers to how well the artifact and its analytical parts accurately represent the decision-making constructs they aim to support. In this study, concepts like energy autonomy, self-consumption, surplus, and uncovered demand are operationalized using indicators from SECAP methodologies and energy-planning literature. Although these constructs are common in policy and planning contexts, they simplify complex socio-technical phenomena into measurable indicators (Trutnevyte et al., 2019). This simplification presents a limitation: the system provides analytical representations of energy performance but does not encompass all real-world factors such as political negotiations, stakeholder conflicts, institutional capacity, or regulatory uncertainty. This aligns with the purpose of planning support systems, which are meant to guide and structure reasoning rather than fully represent every aspect of decision-making (Geertman & Stillwell, 2009). Therefore, the artifact should be seen as a tool for analytical judgment rather than a complete model of planning behavior.

3.9.2 Internal validity

Internal validity determines whether the observed results are truly due to the design logic of the artifact rather than uncontrolled confounding factors. In this study, internal validity is enhanced by clearly defining indicators, enforcing mass-balance constraints, and maintaining consistent aggregation rules across different spatial scales and scenarios. These measures ensure that variations in scenario outcomes are attributable to changes in assumptions, not computational errors. However, internal validity is limited by data uncertainties and model assumptions. Data on energy consumption, renewable production, and spatial suitability may have measurement errors, gaps, or simplifications, as documented in the literature on territorial energy planning and system modeling (Pfenninger et al., 2014). Although the artifact aims for transparency by showing how indicators react to assumptions, it cannot eliminate all uncertainty. Instead, uncertainty becomes part of the decision-making context rather than an error to be corrected.

3.9.3 External validity and generalizability

External validity refers to how well findings and design principles apply beyond the specific context examined. The artifact is explicitly based on European energy-planning frameworks, especially SECAP methodologies and related reporting standards. Thus, its direct relevance to non-European or non-policy-based planning scenarios may be limited. However, this research's contribution is not in the particular institutional setup but in the underlying design principles derived from the study. These include multi-scale consistency, clear surplus representation, and human-centered scenario exploration—concepts that align with broader findings in planning support and decision-support systems research (Arnott & Pervan, 2005; Geertman & Stillwell, 2009). Still, validating these principles in additional territorial and governance contexts is necessary to verify their broader applicability.

3.9.4 Evaluation validity

Evaluation validity concerns whether the evaluation method provides credible evidence that the artifact achieves its intended purpose. This study uses scenario-based analytical evaluation with decision tasks instead of empirical user testing or long-term deployment. This approach is suitable for exploratory decision-support systems focused on assessing reasoning and sense-making support rather than behavioral outcomes (Hevner et al., 2004; Shneiderman et al., 2016).

3.9.5 Epistemic and normative limitations

An additional limitation relates to the artifact's epistemic stance. Scenario-based decision-support systems naturally influence problem perception by highlighting specific indicators, spatial representations, and temporal scales. As noted in the literature on models and policy advice, analytical tools do more than inform—they shape what is considered relevant or legitimate knowledge (Saltelli et al., 2020). In this study, choosing to emphasize transparency, comparability, and non-prescriptiveness is a normative decision aligned with public-sector accountability and SECAP reporting requirements. Although this enhances interpretability, it may limit the system’s ability to support detailed techno-economic optimization or predictive forecasting. This trade-off is intentional and clearly acknowledged. Overall, these validity concerns do not undermine the study's contribution; rather, they clarify its scope. The artifact should be seen as a human-centered planning support system that improves analytical reasoning, enables scenario comparison, and increases transparency in territorial energy planning. It is not designed as a predictive simulator or an automation tool. Recognizing these limitations openly provides a strong basis for future development, empirical validation, and institutional use.

Chapter 4 — System Architecture and Data Model

This chapter demonstrates that the proposed system goes beyond a mere visualization dashboard; it functions as a scientifically based decision-support system (DSS) designed to assist with territorial energy planning across different spatial scales, timeframes, and policy options. It integrates proven planning principles—commonly used in expert tools—into an interactive, easy-to-use, and flexible digital platform.

4.1 Architectural Role of the System as a Decision-Support Tool

From an expert's perspective, a territorial energy dashboard should support semi-structured decision-making, where objectives are clear but optimal solutions depend on interpretation, comparison, and scenario analysis. This positions the system as a decision-support tool rather than just a reporting or monitoring interface. The primary decision-makers include municipal energy officers, regional planners, and technical consultants involved in SECAP development and monitoring. The dashboard supports decisions related to:

1. Spatial prioritization of interventions,
2. Comparing energy performance across territories,
3. Analyzing temporal trends,
4. Evaluating different future scenarios.

Interactivity is crucial here. Unlike static reports or dashboards with fixed indicators, the system enables users to actively explore data through an interactive map, linked charts, and comparison views. These features help reduce cognitive load by translating complex analytical relationships into visual and interactive formats, which is essential for effective decision support systems (Power, 2007; Arnott & Pervan, 2012).

Compared to benchmark platforms—often expert-only or model-centric—the proposed system emphasizes exploratory reasoning, allowing planners to refine questions rather than merely consume pre-defined outputs iteratively.

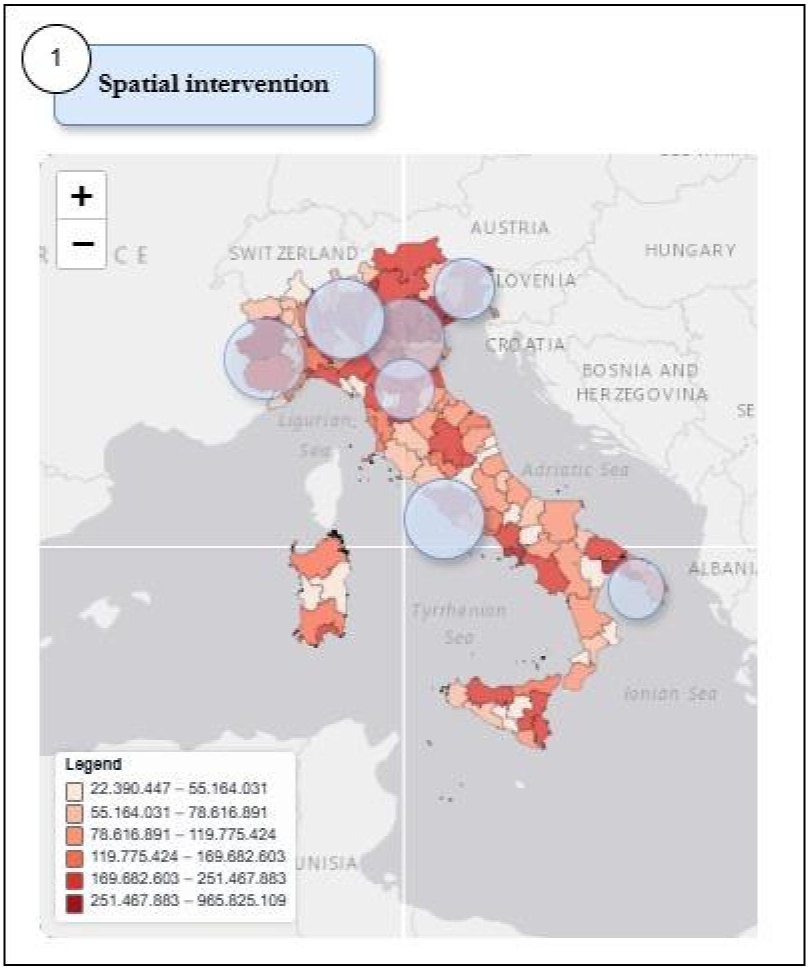

Figure 4.1 — The initial section illustrates how the dashboard functions as a decision-support system by highlighting the prioritization of spatial interventions.

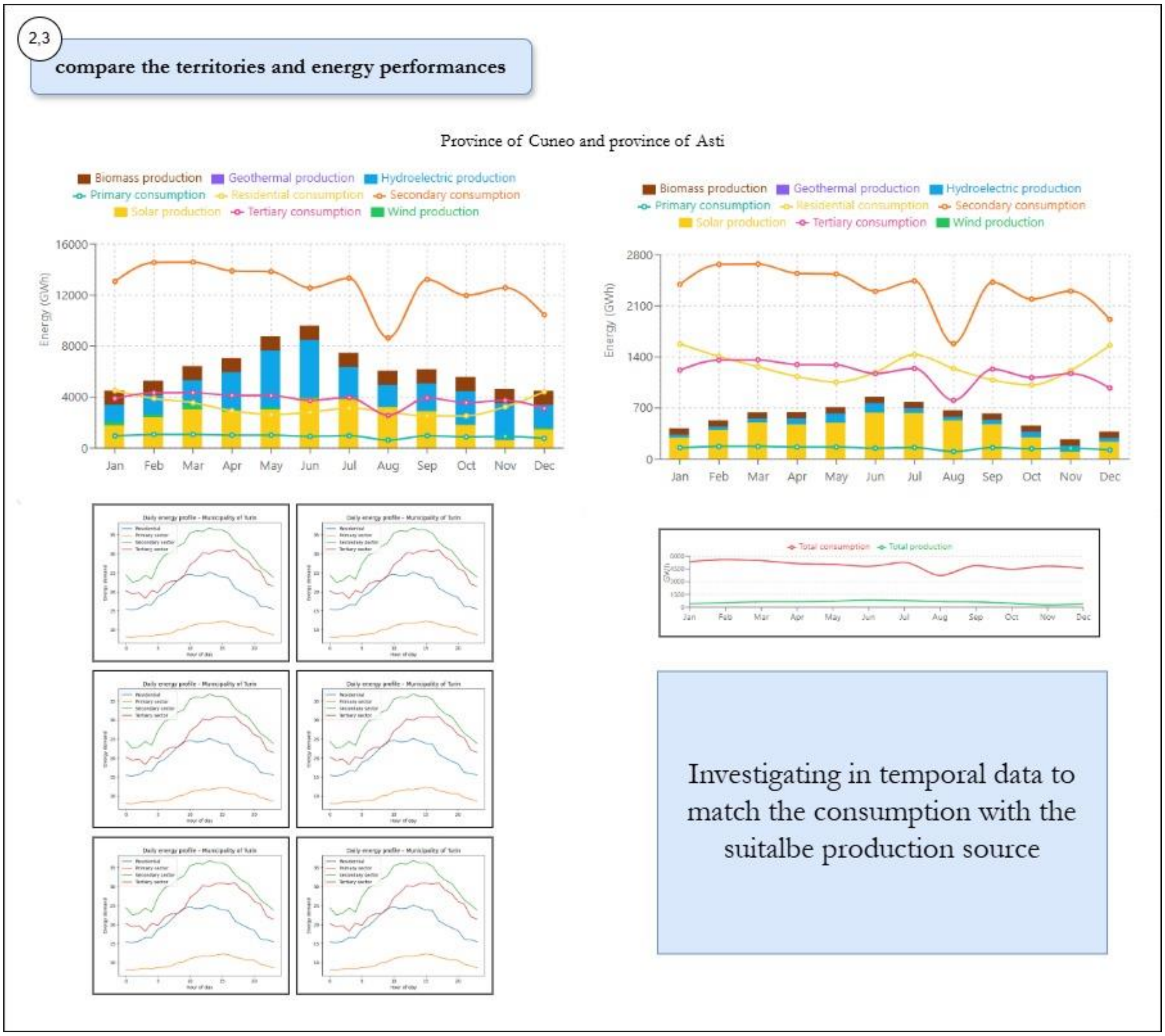

Figure 4.2- Understanding the dashboard's role as a decision-support system, parts 2 and 3: Comparing energy performance across regions and analyzing temporal trends

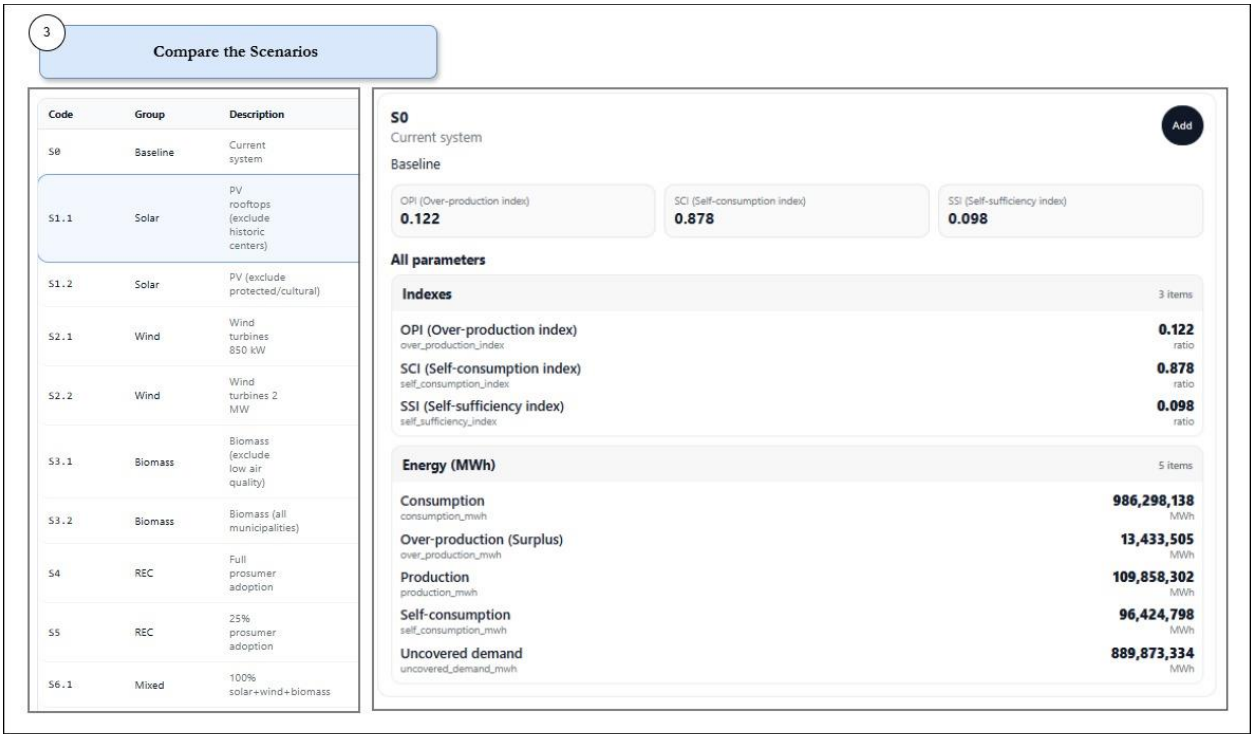

Figure 4.3- Role of the dashboard as a decision-support system: Part 4 — Comparing the scenarios.

4.2 Separation of Concerns and Analytical Responsibility

A crucial aspect of analytical systems is ensuring a distinct separation between data storage, analytical logic, and user interaction. In this system, all indicator calculations and aggregation processes are centralized and operate independently from the dashboard interface. The dashboard does not directly compute indicators; instead, it relies on validated, pre-aggregated analytical results, which guarantee consistent use of the same indicator definitions across:

1. interactive maps
2. interactive charts
3. territorial comparison views
4. Scenario comparisons

This design avoids the common issue found in many lightweight dashboards, where analytical assumptions are implicitly built into the visualization logic. Unlike these, expert platforms like EnergyPLAN use centralized computation; the proposed system maintains this methodological rigor while offering more interactive features. This separation of concerns is crucial for ensuring correctness, reproducibility, and long-term maintenance (March & Smith, 1995; Bass et al., 2012).

4.3 Territorial Modeling for Multi-Level Governance

Territorial energy planning inherently operates on multiple levels. Municipalities, provinces, and regions each have distinct governance roles and cannot be treated as interchangeable spatial units. As a result, the system employs a clear hierarchical territorial model where higher-level indicators are systematically aggregated from lower-level units. This modeling approach directly supports a key analytical feature of the dashboard: the ability to compare territories. Users can pick a municipality, province, or region on the interactive map and analyze its indicators alongside others using the same aggregation method. Unlike ad hoc GIS dashboards that rely on manual spatial joins, this system ensures consistent methodology across scales, aligning with European planning practices and JRC guidelines that emphasize coherence across administrative levels for policy assessment (European Commission, JRC 2018).

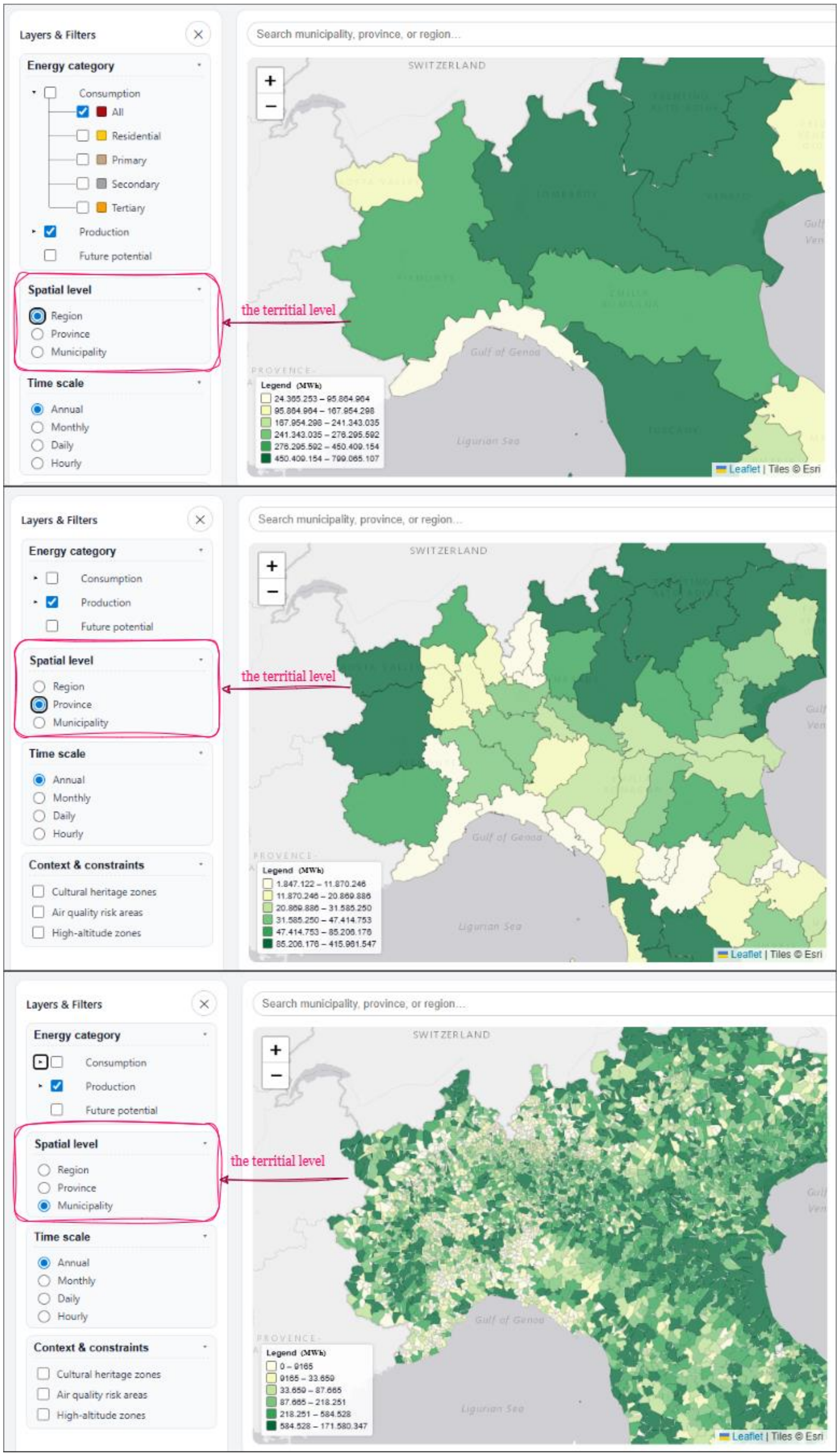

Figure 4.3 — Territorial hierarchy and aggregation logic
(Municipality, Province, Region with deterministic aggregation paths)

4.4 Temporal Modeling and Analytical Flexibility

Energy analysis necessitates explicit control over temporal granularity. The system incorporates multiple temporal resolutions—such as yearly, monthly, hourly, and seasonal—explicitly modeled within the analytical layer rather than inferred at the interface. Interactive charts reveal this temporal flexibility, allowing users to conduct trend analysis, compare seasonal patterns, and align timelines across territories and scenarios. Temporal aggregation is managed centrally to ensure indicators remain consistent across timeframes. This approach combines the analytical rigor of expert modeling tools with a user-friendly interaction style, aligning with visual analytics principles (Kimball & Ross, 2013; Shneiderman et al, 2016).

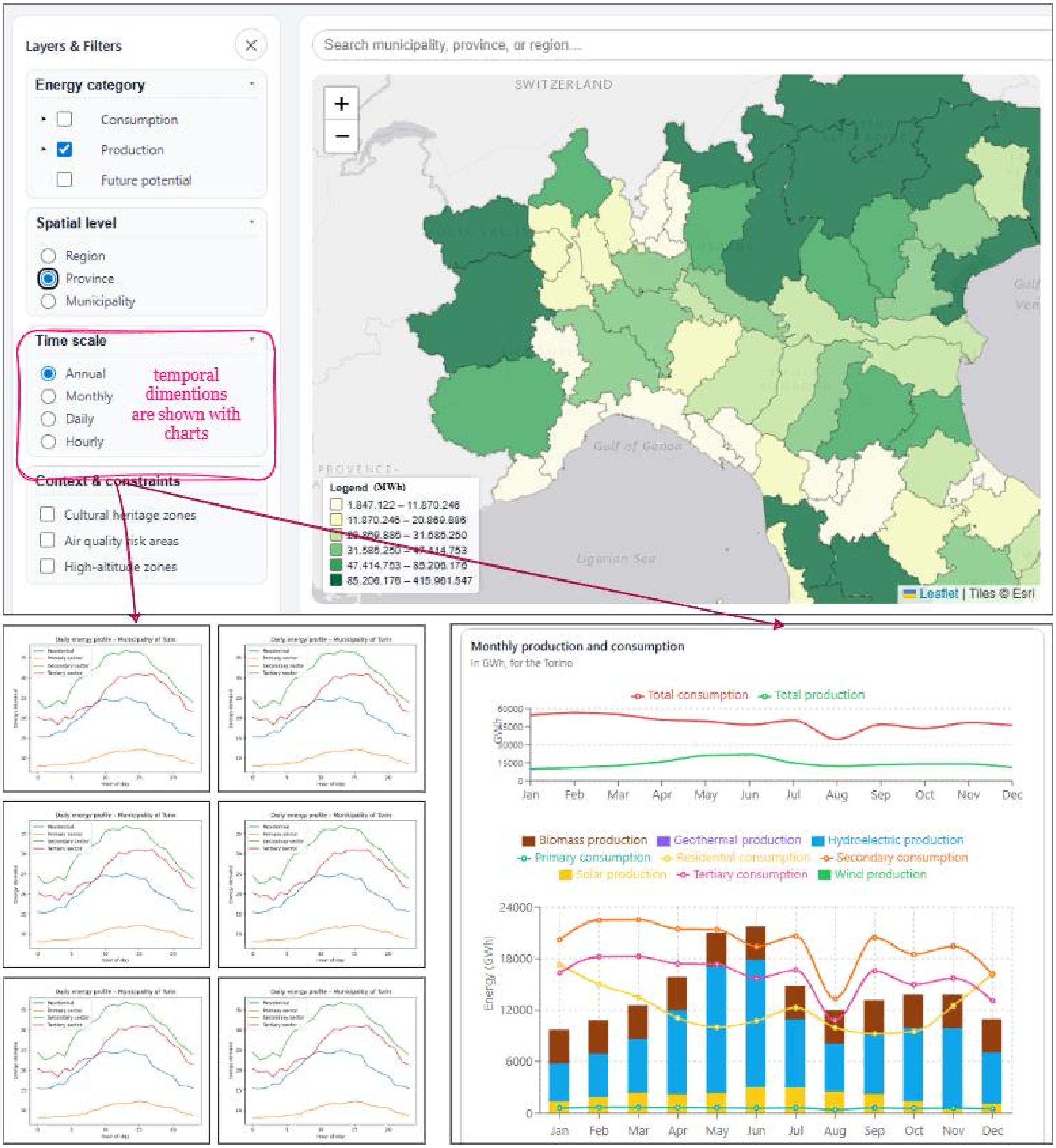

Figure 4.4 — Temporal dimensions and dashboard interaction
(Temporal filters linked to consistent indicator computation)

4.5 Indicator Formalization and Transparency

In expert-level decision-support systems, indicators must be well-defined, transparent, and policy-relevant. The system formalizes these indicators through a structured computation process, keeping it separate from visualization. Energy consumption, renewable production, and planning

Indicators are calculated once and reused across all web application views, ensuring that differences in maps or charts represent real analytical variations rather than visualization artifacts. These indicator definitions align with SECAP reporting standards, enhancing their applicability in actual planning workflows and institutional use (European Commission, JRC, 2018).

SECAP Indicator (Reference)	Dashboard KPI / Indicator	Data Requirements	Visualization & Interaction	Supported Decision Question
Total final energy consumption by sector (BEI/MEI)	Total final energy consumption (MWh)	Energy consumption by sector, territory, time	Choropleth map + time-series chart	Which territories consume the most energy, and in which sectors?
Final energy consumption by sector	Sectoral consumption share (%)	Sector-level consumption data	Stacked bar chart, filterable by scale	Which sectors dominate energy demand at each territorial level?
Renewable energy production by source	Total renewable production (MWh)	Production data by RES type and territory	Map with production layers + grouped bars	Where is renewable energy produced, and from which sources?
Share of renewable energy (%)	Renewable share of final consumption (%)	Consumption + renewable production	Ratio indicator with contextual tooltip	How close is a territory to renewable penetration targets?
CO_2 emissions from final energy consumption	Estimated CO_2 emissions (tCO_2)	Energy consumption + emission factors	Time-series chart + comparative bars	Which territories contribute most to emissions?
Change in energy consumption over time	Energy consumption trend (%)	Multi-year consumption data	Indexed line chart	Are efficiency measures reducing consumption over time?
Local energy balance	Self-sufficiency ratio (%)	Local production + local consumption	Ratio gauge + comparison view	How much local demand can be met by local production?
Local use of RES	Self-consumption ratio (%)	RES production + local use estimates	Scenario-sensitive indicator	How much locally produced energy is consumed locally?
Progress vs baseline	Change relative to BEI (%)	Baseline + monitoring inventory	Difference charts + reference lines	Is the territory progressing in accordance with its SECAP targets?
Policy scenario comparison	Scenario delta (Δ MWh, Δ %)	Scenario-based projections	Side-by-side scenario view	What changes under alternative planning assumptions?
Seasonal variation	Seasonal consumption/production patterns	Sub-annual time resolution	Seasonal line/area charts	When are peaks and mismatches occurring?
Territorial contribution	Contribution to higher-level total (%)	Aggregated values across scales	Ranked lists + drill-down	Which municipalities drive provincial or regional results?

Table 4.1 presents the mapping between indicators commonly used in the SECAP framework and the key performance indicators implemented in the proposed dashboard. In addition to SECAP-aligned indicators, the dashboard includes derived indicators and scenario-based metrics designed to support exploratory analysis and multi-scale comparison, while remaining consistent with the objectives of territorial energy planning.

4.5 Frontend architecture and codebase organization

The frontend of the proposed decision-support platform is organized following a layered, responsibility-driven structure, designed to support analytical complexity while remaining maintainable and extensible over time. The directory organization reflects a deliberate separation between presentation components, application logic, data access, and semantic definitions, translating the architectural principles introduced earlier in this chapter into concrete implementation choices.

The analysis informed this organization of legacy energy applications observed during the internship at CSI Piemonte, where functionality was often embedded in monolithic views and tightly coupled to specific datasets or interaction flows. In contrast, the proposed structure explicitly encodes architectural roles within the codebase, reducing ambiguity and facilitating future development and institutional handover.

4.5.1 Component layer

The components/ directory holds all reusable user interface elements and layout structures. These components are organized by their functional role rather than by technical type, which aids in a clearer understanding of system behavior and user interactions.

Core layout and orchestration components include:

• `Grid.tsx`, which defines the overall layout, coordinating the side panel, map, and chart areas.

• `MapShell.tsx`, which encompasses the cartographic container and related UI controls.

• `ChartShell.tsx`, offering a dedicated space for analytical visualizations that are synchronized with the map.

• `Header.tsx` and `Toolbar.tsx`, which manage global navigation and application controls.

These components are intentionally designed to be independent of specific datasets or indicators, allowing for future analytical extensions. do not require refactoring the layout.

Map-related components

The map/ subdirectory contains cartographic components responsible for spatial visualization and interaction:

• The main choropleth rendering logic is encapsulated in a dedicated map component, which receives geometries and values independently of spatial scale or thematic domain.

• Legend.tsx renders dynamic legends based on classification rules and color palettes.

• PlaceInfo.tsx provides contextual information about selected spatial entities.

This separation ensures that cartographic behavior (styling, interaction, classification) remains reusable across different analytical contexts and planning questions.

Filter and interaction components

Explicit filter components facilitate user-driven analytical setup:

• LayersFilterPanel.tsx features a QGIS-like hierarchical layer tree, revealing thematic domains, energy categories, spatial scales, and temporal resolutions.

• TerritoryLevel.tsx allows direct adjustment of spatial aggregation levels.

• SidePanel.tsx contains filter components and relevant controls.

By making analytical parameters visible in the UI, the system enhances transparency and interpretability—essential for decision-support tools used in public-sector planning.

Charts, pages, and auxiliary components

The charts/ directory includes visualization components connected to the map's analytical state.

• The pages/ directory organizes higher-level views made up of multiple components.

• Auxiliary components like DownloadReportButton.tsx facilitate reporting and export features.

• Administrative and configuration components are housed in the admin/ directory, ensuring the analytical logic remains separate from management functions.

4.5.2 Contexts and application state

The 'contexts/' directory contains shared React Contexts that manage application-wide state, including spatial scale, thematic domain, temporal resolution, active overlays, and navigation status. Centralizing these parameters provides a single source of truth for configuration, ensuring all components respond uniformly to changes in user intent. This approach prevents inconsistencies between maps, charts, and filters, addressing a common issue found in traditional dashboard systems.

4.5.3 Data access layer (custom hooks)

The hooks/ directory contains all data fetching and synchronization functions implemented as custom React hooks. Each hook targets a specific analytical requirement and aggregation level, such as:

- useDailyData.ts
- useMonthlyData.ts
- useHourlyData.ts
- useHourlyCalendarData.ts

These hooks hide the complexities of backend interactions from the presentation components, providing normalized data structures ready for visualization. This setup allows backend changes in aggregation or API structure to be managed within the hooks, leaving the UI unaffected. The multiple hooks indicate deliberate analytic granularity, not duplicated logic, with each hook representing a distinct spatial–temporal query that enhances code clarity and analytical tracing.

4.5.4 Type definitions and semantic consistency

The 'types/' directory centralizes shared type definitions used across the application, including spatial entity identifiers, analytical domains, and tabs (e.g., AppTab.ts), filter state representations, and API response structures. By centralizing all semantic definitions in a single place, the architecture minimizes the risk of discrepancies between frontend expectations and backend outputs. This is particularly critical in multi-scale energy planning tools, where even minor semantic mismatches can lead to misinterpretation of results.

4.5.5 Utilities and analytical helpers

Reusable, UI-agnostic logic is organized in the utils/ directory. These utilities offer helper functions for:

- normalizing and parsing values,
- resolving spatial identifiers,
- generating classifications and color scales,
- mapping parameters between frontend state and backend queries.

Separating this logic from the components ensures that analytical assumptions remain clear and accessible, thereby preventing the hidden dependencies that are common in legacy systems with extensive visualization.

4.5.6 Architectural implications for sustainability and future development

The chosen frontend organization directly tackles the limitations found in current CSI Piemonte energy applications. By embedding architectural responsibilities within the directory structure, the system facilitates the gradual addition of new datasets and indicators, supporting future expansion and adaptability. This design reassures users about the platform's long-term sustainability and scalability. It also allows the platform to adapt to evolving policy requirements, data sources, and planning practices without requiring disruptive overhauls.

Table 4.3 — Frontend directory structure and architectural responsibility

Directory	Main content	Architectural role
components/	UI elements, layout shells, maps, filters, charts	Presentation layer
components/contexts/	Shared application state	State coordination
hooks/	Data retrieval and aggregation logic	Application logic
types/	Shared semantic definitions	Consistency layer
utils/	Reusable analytical helpers	Support layer
assets/	Icons and visual resources	Visual identity
Root files	App bootstrap and configuration	System integration

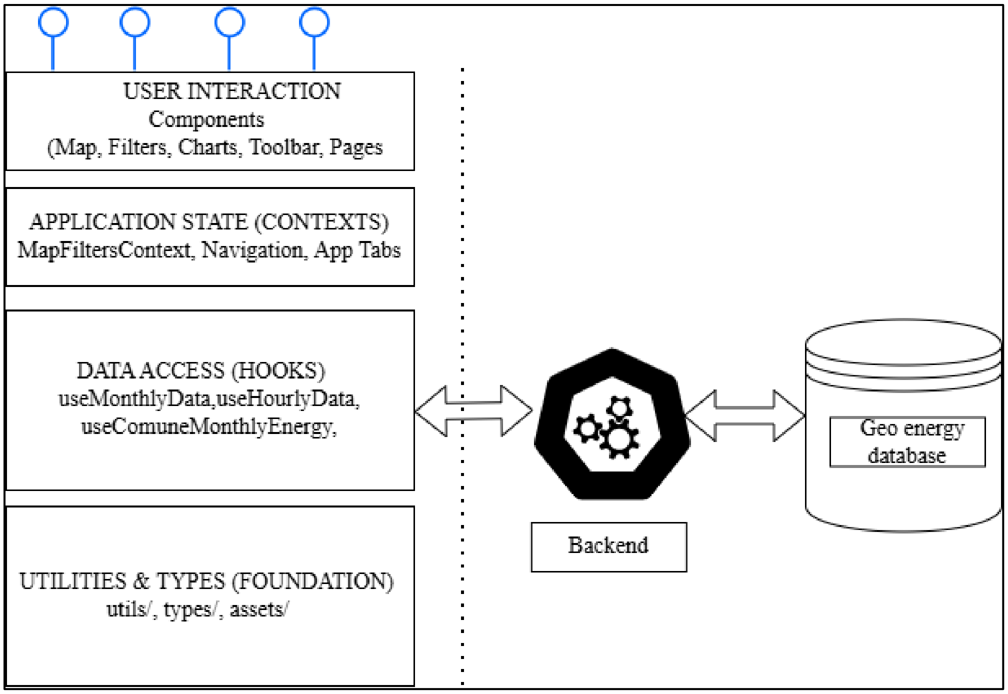

Figure 5.4: The diagram of the work done is on the left part

Chapter 5 — Interface Design and Implementation

This chapter describes how the earlier discussed analytical capabilities are integrated into a practical interface for real-world planning. The dashboard is viewed not as a simple visualization tool but as an active decision-support component, structured around the actual tasks of energy planners. Users — municipal energy officers, regional planners, and technical consultants — need to compare regions, interpret indicators, and explore alternatives flexibly, without strict workflows. Therefore, the interface is designed for easy navigation across different spatial levels, time periods, and scenarios. Users can seamlessly switch between regional, provincial, and municipal views to analyze trends or test assumptions. Interaction supports exploratory thinking: each action gives immediate feedback, filters are visible, and operations can be reversed. This enables analysts to test hypotheses, modify parameters, and see results without losing context. The goal isn't to automate decisions but to create a stable analytical environment that fosters human judgment, comparison, and iterative reasoning.

The visual design of the interface supports this exploratory approach. The map serves as the main analytical surface, facilitating spatial comparisons and pattern recognition, while charts display temporal and sectoral details linked directly to map selections. Consistent color schemes and legends across scales ensure that switching between aggregation levels does not alter the meaning of visual cues. This consistency helps prevent misinterpretation when users move between different territorial levels. Scenario analysis is integrated into the same visual framework rather than being a separate tool: baseline and alternative assumptions update maps and charts immediately, enabling quick "what-if" assessments. Technically, the dashboard is built as a modular web system where components share state, ensuring synchronized updates without skewing analysis. Performance improvements are only applied when they do not impact indicator accuracy. The resulting interface functions less like a static report and more like an interactive workspace, where spatial reasoning, temporal analysis, and scenario testing operate together in a single, coordinated environment.

Interface component	Supported task
Interactive map	Spatial comparison and selection
Filters panel	Analytical focus refinement
Charts	Temporal and sectoral analysis
Scenario controls	What-if exploration
Export/report view	Communication and reporting

Table 5.1 — Mapping between interface components and supported decision tasks.

5.1 Technical contribution during the internship at CSI Piemonte

During the internship at CSI Piemonte, the focus was on analyzing the architecture and constraints of existing energy web applications. The work involved gradually redesigning interface elements to create a sustainable, modular, and scalable frontend. The initial review of the legacy system uncovered typical problems seen in long-standing public-sector applications: dependency rigidity, unclear component boundaries, and low reusability of components for new analytical features such as multi-scale exploration, scenario comparison, and interactive filtering.

5.8.1 Legacy applications and motivation for modernization

The legacy energy applications used during the internship were functional but difficult to expand due to several issues:

- Strong coupling between UI elements and specific data formats/endpoints.

- Fragility of versions, where updates to mapping or UI libraries could potentially disrupt core functions.

- Limited modularity, which made adding new territorial scales, layers, or thematic views challenging without duplicating code.

From a sustainability standpoint—focusing on software sustainability rather than just energy efficiency—the internship aimed to develop a frontend architecture that is:

- Component-based, allowing new features to be built as reusable blocks instead of entire pages.

- Configuration-driven, facilitating thematic changes such as switching palettes between consumption and production or adjusting time aggregation without rewriting rendering logic.

- Stable amidst dependency updates by encapsulating external libraries with small wrappers and maintaining application logic within controlled components and hooks.

5.2 Frontend strategy: React & JavaScript for maintainable components

To mitigate long-term risks of breaking changes and facilitate ongoing development, the frontend employs React with a component-first approach, emphasizing:

- Encapsulating mapping logic within a dedicated MainMap component.

- Maintaining a single source of truth for filters through a shared MapFiltersContext (covering theme, scale, time resolution, overlays).

- Differentiating between shells and content to preserve layout stability during updates to the map and analytical views.

A primary design decision was to develop React components using JavaScript. This choice aims to maximize compatibility and minimize friction, especially in environments where strict typing is difficult to enforce across various teams or evolving packages. This approach promotes long-term sustainability by:

- Keeping core logic in small utility functions (e.g., quantiles, value parsing),

- Isolating data-fetching logic in hooks (e.g., useGeoData),

- And avoiding reliance on unstable or experimental third-party APIs libraries.

5.3 Choropleth mapping component as a reusable "decision-support primitive."

A key artifact created during the internship is a choropleth map component that enables:

- Multi-scale rendering (region, province, municipality)

- Multi-domain theming (consumption, production, future potential)

- Multi-resolution time aggregation (annual, monthly, daily, hourly).

The main technical innovation is that the map's behavior is not tied to a single dataset. Instead, it dynamically reads the user's current filter settings and translates them into backend parameters such as level, domain, and resolution. This design allows the same component to be reused across different contexts within the platform.

Code Listing 5.1 — Choropleth Map with Related Functions

```
// src/components/maps/MainMap.tsx
import { useMemo } from "react";
import { MapContainer, TileLayer, GeoJSON } from "react-leaflet";
import "leaflet/dist/leaflet.css";
import type { Feature, Geometry } from "geojson";

import Legend from "../Legend";
import { useGeoData } from "./hooks/useGeoData";
import { useMapFilters } from "../contexts/MapFiltersContext";
import {
  scaleToBackendLevel,
  themeToBackendDomain,
  timeResolutionToBackendResolution,
  type BackendLevel,
} from "./hooks/mapFiltersToGeoArgs";

// ---------- Minimal helpers ----------
const  CONSUMPTION_PALETTE  =  ["#FEE5D9",  "#FCBBA1",  "#FC9272",
"#FB6A4A", "#DE2D26", "#A50F15"];
const  PRODUCTION_PALETTE  =  ["#FFFFE5",  "#F7FCB9",  "#C2E699",
"#78C679", "#31A354", "#006837"];

function toNum(x: unknown): number | null {
```

```
  if (typeof x === "number" && Number.isFinite(x)) return x;
  if (typeof x === "string") {
    const n = Number(x.trim());
    return Number.isFinite(n) ? n: null;
  }
  return null;
}

function quantileBreaks(values: number[], classes: number) {
  const sorted = [...values].filter((v) => Number.isFinite(v)).sort((a, b) => a - b);
  if (sorted.length === 0) return [];
  const n = sorted.length;
  const b: number[] = [];
  for (let i = 0; i <= classes; i++) {
    const idx = Math.floor((i / classes) * (n - 1));
    b.push(sorted[idx]);
  }
  return b;
}

function colorForValue(v: number | null | undefined, breaks: number[], colors: string[]) {
  if (!Number.isFinite(v as number) || breaks.length < 2) return "#ccc";
  const val = v as number;

  for (let i = 0; i < breaks.length - 1; i++) {
    const from = breaks[i];
    const to = breaks[i + 1];

    if (i === breaks. length - 2) {
      if (val >= from && val <= to) return colors[i] ?? colors[colors.length - 1];
    } else if (val >= from && val < to) {
      return colors[i] ?? colors[colors.length - 1];
    }
  }
  return colors[colors.length - 1];
}

// ---- Code extractors ----
function getRegionCode(p: any): number | null {
  return toNum(p?.COD_REG) ?? toNum(p?.cod_reg) ?? toNum(p?.REG_COD) ?? toNum(p?.reg_cod) ?? toNum(p?.id_reg) ?? toNum(p?.id) ?? null;
}
function getProvinceCode(p: any): number | null {
```

```
  return    toNum(p?.COD_PROV)    ??    toNum(p?.cod_prov)    ??
toNum(p?.PROV_COD) ?? toNum(p?.prov_cod) ?? toNum(p?.COD_UTS) ??
toNum(p?.cod_uts) ?? toNum(p?.id_prov) ?? toNum(p?.id) ?? null;
}
function getComuneCode(p: any): number | null {
  return    toNum(p?.PRO_COM)    ??    toNum(p?.pro_com)    ??
toNum(p?.MUN_COD) ?? toNum(p?.mun_cod) ?? toNum(p?.COD_COM) ??
toNum(p?.cod_com)  ??  toNum(p?.ISTAT)  ??  toNum(p?.istat)  ??
toNum(p?.id_mun) ?? toNum(p?.id) ?? null;
}

function codeForFeature(level: BackendLevel, p: any): number | null
{
  if (level === "region") return getRegionCode(p);
  if (level === "province") return getProvinceCode(p);
  return getComuneCode(p);
}

function nameFor(level: BackendLevel, p: any): string {
  if (level === "region") return (p?.DEN_REG ?? p?.den_reg ??
p?.name ?? "").toString();
  if (level === "province") return (p?.DEN_UTS ?? p?.den_uts ??
p?.name ?? "").toString();
  return (p?.DEN_COM ?? p?.den_com ?? p?.name ?? "").toString();
}

export default function MainMap() {
  const { filters } = useMapFilters();

  const level = useMemo(() => scaleToBackendLevel(filters.scale),
[filters.scale]);
  const        domain        =        useMemo(()        =>
themeToBackendDomain(filters.theme), [filters.theme]);
  const palette = useMemo(() => {
    if (domain === "production" || domain === "future_production")
return PRODUCTION_PALETTE;
    return CONSUMPTION_PALETTE;
  }, [domain]);

  const resolution = useMemo(
    ()                                                        =>
timeResolutionToBackendResolution(filters.timeResolution),
    [filters.timeResolution]
  );

  const year = 2019;
  const scenario = 0;
```

```
  const baseGroup = useMemo(() => filters.consumptionBaseGroup,
[filters.consumptionBaseGroup]);

  const { geo, valuesMap, loadingGeo } = useGeoData({
    level,
    domain,
    resolution,
    year,
    scenario,
    baseGroup,
  });

  const breaks = useMemo(() => {
    const vals = Array.from(valuesMap.values());
    return quantileBreaks(vals, palette.length);
  }, [valuesMap, palette]);

  const style = (feature: any) => {
    const p = feature?.properties ?? {};
    const code = codeForFeature(level, p);
    const v = code != null ? valuesMap.get(code) : undefined;

    return {
      color: "rgba(0,0,0,0.25)",
      weight: level === "comune" ? 0.35 : 1,
      fillOpacity: 0.75,
      fillColor: colorForValue(v, breaks, palette),
    };
  };

  const onEachFeature = (feature: Feature<Geometry, any>, layer:
any) => {
    const p = feature?.properties ?? {};
    const nm = nameFor(level, p);
    const code = codeForFeature(level, p);
    const val = code != null ? valuesMap.get(code) : undefined;

    layer.on("click", () => {
      layer.bindPopup(
        `<div style="font-size:12px">
          <div><b>${nm || "—"}</b></div>
          <div>code: ${code ?? "—"}</div>
          <div>value: ${val ?? "—"}</div>
        </div>`
      );
      layer.openPopup();
    });
  };
```

```
  return (
    <div style={{ height: "100%", width: "100%", position: "relative" }}>
      {loadingGeo && (
        <div style={{
          position: "absolute", inset: 0, zIndex: 1500, display: "flex",
          alignItems: "center", justifyContent: "center",
          background: "rgba(255,255,255,0.75)", fontSize: 14, pointerEvents: "none",
        }}>
          Loading…
        </div>
      )}

      <MapContainer center={[41.9, 12.5]} zoom={6} style={{ height: "100%", width: "100%" }} preferCanvas>
        <TileLayer

url="https://server.arcgisonline.com/ArcGIS/rest/services/Canvas/World_Light_Gray_Base/MapServer/tile/{z}/{y}/{x}"
          attribution="Tiles © Esri"
        />

        {geo && (
          <GeoJSON
            key={`${level}-${domain}-${resolution}`}
            data={geo as any}
            style={style as any}
            onEachFeature={onEachFeature as any}
          />
        )}

        {breaks.length > 1 && <Legend breaks={breaks} colors={palette} />}
      </MapContainer>
    </div>
  );
}
```

This component formalizes the choropleth as a consistent “decision-support primitive”: the same rendering logic allows for changing scale, theme, and time aggregation, enabling future extensions without rewriting the map.

5.8.4 Layout sustainability: shells and grid composition

To ensure extensibility, the UI is organized into stable layout shells (grid, panel, map shell, chart shell) and replaceable content components. This avoids the need for layout refactoring every time a new layer type or chart is added.

Example: Grid composition

Code Listing 5.2 — Grid layout with side panel toggle and stable shells

```
// src/components/Grid.tsx
import { useState, type ReactNode } from "react";
import "../index.css";

import SidePanel from "./SidePanel";
import MapShell from "./MapShell";
import ChartShell from "./ChartShell";

type GridProps = {
  map: ReactNode;
  side?: ReactNode;
};

export default function Grid({ map, side }: GridProps) {
  const [panelOpen, setPanelOpen] = useState(true);

  const togglePanel = () => setPanelOpen((v) => !v);

  return (
    <>
      {panelOpen && (
        <SidePanel onClose={togglePanel}>
          {side ?? <p>Side panel content…</p>}
        </SidePanel>
      )}

      <section className="content">
        <MapShell map={map} onTogglePanel={togglePanel} />
        <ChartShell />
      </section>
    </>
  );
}
```

5.8.5 Filter architecture: QGIS-like layer tree for transparent decision interaction

In decision-support dashboards, interpretability improves when users can see which data are active and how they are grouped. The layer/filter panel was therefore designed as a QGIS-like tree (categories, subcategories, and options), as shown in the figure below. This design makes dashboard operations auditable: users can always see which "view of the system" they are analyzing.

Example: Layer & Filters Panel

Code Listing 5.8 — QGIS-like layer tree and filter controls (energy category, scale, time resolution, overlays)

```
// src/components/LayersFiltersPanel.tsx
import { useEffect, useState, type ReactNode } from "react";
import { useMapFilters, type SpatialScale } from
"./contexts/MapFiltersContext";

import {
  type ProductionType,
  type ConsumptionSector,
  CONSUMPTION_SWATCH,
  PRODUCTION_SWATCH,
  CONSUMPTION_LABEL,
  PRODUCTION_LABEL,
  PRODUCTION_ICON,
  makeSequentialFromBase,
} from "./LayersFilters.assets";

type SectionProps = {
  title: string;
  info?: string;
  defaultOpen?: boolean;
  children: ReactNode;
};

function Section({ title, info, defaultOpen = true, children }:
SectionProps) {
  const [open, setOpen] = useState(defaultOpen);

  return (
    <div className="side-section">
      <button
        type="button"
        className="side-section-header"
        onClick={() => setOpen((o) => !o)}
      >
        <div className="side-section-title-wrap">
```

```
          <span className="side-section-title">{title}</span>
          {info        &&        <span        className="side-section-
subtitle">{info}</span>}
        </div>
        <span    className="side-section-toggle">{open    ?    "▾"    :
"▸"}</span>
      </button>

      {open && <div className="side-section-body">{children}</div>}
    </div>
  );
}

/** QGIS-like tree node with connectors + checkbox + optional icon +
swatch + children */
type TreeNodeProps = {
  label: string;
  checked?: boolean;
  onCheck?: () => void;
  defaultOpen?: boolean;
  swatchColor?: string | null;
  icon?: ReactNode;
  isLast?: boolean;
  children?: ReactNode;
};

function TreeNode({
  label,
  checked = false,
  onCheck,
  defaultOpen = true,
  swatchColor,
  icon,
  isLast = false,
  children,
}: TreeNodeProps) {
  const hasChildren = Boolean(children);
  const [open, setOpen] = useState(defaultOpen);

  useEffect(() => {
    if (hasChildren && checked) setOpen(true);
  }, [checked, hasChildren]);

  const nodeClassName = [
    "qgis-node",
```

```
    isLast ? "is-last" : "",
    hasChildren && open ? "has-open-children" : "",
  ]
    .filter(Boolean)
    .join(" ");

  return (
    <div className={nodeClassName}>
      <div className="qgis-row">
        {hasChildren ? (
          <button
            type="button"
            className="qgis-toggle"
            onClick={() => setOpen((o) => !o)}
            aria-label={open ? "Collapse" : "Expand"}
          >
            {open ? "▾" : "▸"}
          </button>
        ) : (
          <span className="qgis-toggle-placeholder" />
        )}

        <span className="qgis-check-anchor">
          <input         type="checkbox"         checked={checked}
onChange={onCheck} />
        </span>

        {icon ? <span className="qgis-icon">{icon}</span> : null}

        {swatchColor ? (
          <span   className="qgis-swatch"   style={{   background:
swatchColor }} />
        ) : (
          <span className="qgis-swatch qgis-swatch--empty" />
        )}

        <span className="qgis-label">{label}</span>
      </div>

      {hasChildren    &&    open    &&    <div    className="qgis-
children">{children}</div>}
    </div>
  );
}

function  ConsumptionSequentialLegend({  baseColor,  title  }:  {
baseColor: string; title: string }) {
  const colors = makeSequentialFromBase(baseColor);
```

```
  return (
    <div className="consumption-seq">
      <div className="consumption-seq-title">{title}</div>
      <div className="consumption-seq-bar">
        {colors.map((c) => (
          <span key={c} className="consumption-seq-step" style={{
background: c }} />
        ))}
      </div>
      <div     className="consumption-seq-hint">Low     →     High
intensity</div>
    </div>
  );
}

export default function LayersFiltersPanel() {
  const {
    filters,
    setTheme,
    setScale,
    setTimeResolution,
    toggleOverlay,
    setConsumptionBaseGroup,
  } = useMapFilters();

  const        [productionType,        setProductionType]        =
useState<ProductionType>("all");
  const      [consumptionSector,      setConsumptionSector]      =
useState<ConsumptionSector>("all");

  useEffect(() => {
    if (filters.theme === "production") setProductionType("all");
    if          (filters.theme          ===          "consumption")
setConsumptionSector("all");
  }, [filters.theme]);

  const            handleScaleChange            =            (e:
React.ChangeEvent<HTMLInputElement>) => {
    setScale(e.target.value as SpatialScale);
  };

  const selectConsumption = (v: ConsumptionSector) => {
    setConsumptionSector(v);

    if (v === "all") setConsumptionBaseGroup(null);
    else         if         (v         ===         "residential")
setConsumptionBaseGroup("domestic");
```

```
    else  setConsumptionBaseGroup(v  as  "primary"  |  "secondary"  |
"tertiary");
  };

  const     selectProduction     =     (v:     ProductionType)     =>
setProductionType(v);

  return (
    <div className="layers-filters-panel">
      <Section
        title="Energy category"
        info="Choose whether to analyse consumption, production or
future potential."
      >
        <div className="qgis-tree">
          <TreeNode
            label="Consumption"
            checked={filters.theme === "consumption"}
            onCheck={() => setTheme("consumption")}
            defaultOpen={true}
          >
            <TreeNode
              label={CONSUMPTION_LABEL.all}
              checked={consumptionSector === "all"}
              onCheck={() => selectConsumption("all")}
              swatchColor={CONSUMPTION_SWATCH.all}
            />
            <TreeNode
              label={CONSUMPTION_LABEL.residential}
              checked={consumptionSector === "residential"}
              onCheck={() => selectConsumption("residential")}
              swatchColor={CONSUMPTION_SWATCH.residential}
            />
            <TreeNode
              label={CONSUMPTION_LABEL.primary}
              checked={consumptionSector === "primary"}
              onCheck={() => selectConsumption("primary")}
              swatchColor={CONSUMPTION_SWATCH.primary}
            />
            <TreeNode
              label={CONSUMPTION_LABEL.secondary}
              checked={consumptionSector === "secondary"}
              onCheck={() => selectConsumption("secondary")}
              swatchColor={CONSUMPTION_SWATCH.secondary}
            />
            <TreeNode
              label={CONSUMPTION_LABEL.tertiary}
              checked={consumptionSector === "tertiary"}
```

```
              onCheck={() => selectConsumption("tertiary")}
              swatchColor={CONSUMPTION_SWATCH.tertiary}
              isLast
            />

            {filters.theme === "consumption" &&
              consumptionSector !== "all" &&
              CONSUMPTION_SWATCH[consumptionSector] && (
                <ConsumptionSequentialLegend

baseColor={CONSUMPTION_SWATCH[consumptionSector]!}
                  title={`Scale                                for
${CONSUMPTION_LABEL[consumptionSector]}`}
                />
              )}
          </TreeNode>

          <TreeNode
            label="Production"
            checked={filters.theme === "production"}
            onCheck={() => setTheme("production")}
            defaultOpen={true}
          >
            <TreeNode
              label={PRODUCTION_LABEL.all}
              checked={productionType === "all"}
              onCheck={() => selectProduction("all")}
              swatchColor={PRODUCTION_SWATCH.all}
              icon={PRODUCTION_ICON.all}
            />
            <TreeNode
              label={PRODUCTION_LABEL.solar}
              checked={productionType === "solar"}
              onCheck={() => selectProduction("solar")}
              swatchColor={PRODUCTION_SWATCH.solar}
              icon={PRODUCTION_ICON.solar}
            />
            <TreeNode
              label={PRODUCTION_LABEL.wind}
              checked={productionType === "wind"}
              onCheck={() => selectProduction("wind")}
              swatchColor={PRODUCTION_SWATCH.wind}
              icon={PRODUCTION_ICON.wind}
            />
            <TreeNode
              label={PRODUCTION_LABEL.hydroelectric}
              checked={productionType === "hydroelectric"}
              onCheck={() => selectProduction("hydroelectric")}
```

```
              swatchColor={PRODUCTION_SWATCH.hydroelectric}
              icon={PRODUCTION_ICON.hydroelectric}
            />
            <TreeNode
              label={PRODUCTION_LABEL.geothermal}
              checked={productionType === "geothermal"}
              onCheck={() => selectProduction("geothermal")}
              swatchColor={PRODUCTION_SWATCH.geothermal}
              icon={PRODUCTION_ICON.geothermal}
            />
            <TreeNode
              label={PRODUCTION_LABEL.biomass}
              checked={productionType === "biomass"}
              onCheck={() => selectProduction("biomass")}
              swatchColor={PRODUCTION_SWATCH.biomass}
              icon={PRODUCTION_ICON.biomass}
              isLast
            />
          </TreeNode>

          <TreeNode
            label="Future potential"
            checked={filters.theme === "future_potential"}
            onCheck={() => setTheme("future_potential")}
            defaultOpen={false}
            isLast
          />
        </div>
      </Section>

      <Section title="Spatial level" info="Select the territorial
aggregation: region, province or municipality.">
        <div className="side-option-group">
          <label className="side-option">
            <input   type="radio"   name="scale"   value="region"
checked={filters.scale === "region"} onChange={handleScaleChange}
/>
            <span>Region</span>
          </label>

          <label className="side-option">
            <input  type="radio"  name="scale"  value="province"
checked={filters.scale              ===               "province"}
onChange={handleScaleChange} />
            <span>Province</span>
          </label>

          <label className="side-option">
```

```
            <input type="radio" name="scale" value="municipality"
checked={filters.scale            ===            "municipality"}
onChange={handleScaleChange} />
            <span>Municipality</span>
          </label>
        </div>
      </Section>

      <Section title="Time scale" info="Control how the data is
aggregated in time (annual, monthly, daily, hourly).">
        <div className="side-option-group">
          <label className="side-option">
            <input       type="radio"       name="timeResolution"
checked={filters.timeResolution  ===  "annual"}  onChange={()  =>
setTimeResolution("annual")} />
            <span>Annual</span>
          </label>

          <label className="side-option">
            <input       type="radio"       name="timeResolution"
checked={filters.timeResolution  ===  "monthly"}  onChange={()  =>
setTimeResolution("monthly")} />
            <span>Monthly</span>
          </label>

          <label className="side-option">
            <input       type="radio"       name="timeResolution"
checked={filters.timeResolution  ===  "daily"}  onChange={()  =>
setTimeResolution("daily")} />
            <span>Daily</span>
          </label>

          <label className="side-option">
            <input       type="radio"       name="timeResolution"
checked={filters.timeResolution  ===  "hourly"}  onChange={()  =>
setTimeResolution("hourly")} />
            <span>Hourly</span>
          </label>
        </div>
      </Section>

      <Section   title="Context   &   constraints"   info="Toggle
additional map overlays that highlight constraints or sensitive
areas.">
        <div className="side-option-group">
          <label className="side-option">
```

```
            <input                                type="checkbox"
checked={filters.overlays.includes("heritage")}  onChange={()  =>
toggleOverlay("heritage")} />
            <span>Cultural heritage zones</span>
          </label>

          <label className="side-option">
            <input                                type="checkbox"
checked={filters.overlays.includes("air_quality")}   onChange={()
=> toggleOverlay("air_quality")} />
            <span>Air quality risk areas</span>
          </label>

          <label className="side-option">
            <input                                type="checkbox"
checked={filters.overlays.includes("high_altitude")} onChange={()
=> toggleOverlay("high_altitude")} />
            <span>High-altitude zones</span>
          </label>
        </div>
      </Section>
    </div>
  );
}
```

This component is written once and can be reused across different subcategories, preserving the same functionality.

5.8.7 Packages and dependencies adopted for a sustainable frontend architecture

During the internship, the frontend was built using a modern yet cautious selection of libraries to ensure technical strength, ease of long-term maintenance, and sustainability within the organization. The main framework chosen is React, due to its component-based architecture, reliable ecosystem, and widespread use in both industry and public projects. React enables a clear separation among layout, interaction logic, and data visualization components, which decreases coupling and makes future updates easier. For interactive maps, React-Leaflet was used, providing a React interface for the Leaflet mapping library. This guarantees compatibility with established GIS standards without relying on experimental or proprietary rendering systems. Vector geometries are stored using the GeoJSON format, a clear and compatible data standard supported by many GIS tools, spatial databases, and open-data portals. The synchronization of application state—covering spatial scale, thematic areas, time aggregation, and overlays—is managed with lightweight React Contexts and custom hooks. This method avoids dependency-heavy state management tools while ensuring a single, reliable source of truth for key parameters. Additional utility functions, such as quantile classification, color palette assignment, and ID normalization, are developed internally to prevent reliance on external libraries and keep control over analytical processes.

Overall, the selected packages emphasize stability over innovation, favoring mature libraries with strong backward compatibility. This strategy aligns with the needs of public-sector digital platforms, where software longevity, safe upgrades, and institutional handover are key success factors.

5.8.8 Comparison between legacy CSI Piemonte energy applications and the proposed architecture

The table below highlights the key differences between the legacy energy applications at CSI Piemonte and the architecture that was proposed and partially implemented during the internship.

Table 5.3 — Comparison between legacy CSI Piemonte applications and the proposed frontend architecture

Aspect	Legacy CSI Piemonte applications	Proposed architecture (this work)
Frontend paradigm	Page-oriented, tightly coupled views	Component-based React architecture
UI extensibility	New features require new pages or duplicated code	New features composed by reusing existing components
Mapping integration	Map logic embedded directly in pages	Dedicated, reusable map component (MainMap)
Spatial scale handling	Hard-coded per view (region/province/comune)	Unified multi-scale logic driven by filter state
Thematic domains	Separate implementations for consumption/production	Single rendering pipeline with domain abstraction
Temporal aggregation	Fixed or backend-dependent	Explicit user-controlled resolution (annual–hourly)
State management	Implicit or scattered across UI logic	Centralized filter state via React Context
Dependency evolution	High risk of breakage during upgrades	Isolated dependencies, minimal API surface
GIS interoperability	Custom or implicit formats	Standard GeoJSON throughout
Long-term maintainability	Knowledge concentrated in specific implementations	Knowledge encoded in reusable components and hooks
Suitability for future development	Limited, incremental fixes	High: ready for new indicators, scenarios, and layers

Chapter 6 — Evaluation and Results

This chapter assesses the dashboard as a Design Science Research (DSR) artifact. In DSR, evaluation must show that the artifact is useful for its intended decision-making context and that its design meets the necessary qualities (utility, correctness, usability, and robustness). The evaluation approach follows the FEDS framework, which organizes evaluation around why, what, how, and when the artifact is assessed. Because the artifact is an interactive visual analytics dashboard (map + charts + scenario comparison), the evaluation must go beyond interface "usability" and include verification that (i) analytical outputs are accurate and consistent, (ii) visual encodings enable valid interpretation, and (iii) scenario behavior remains coherent under parameter changes. This multi-layered evaluation is formalized using Munzner's Nested Model, which explicitly links evaluation methods to the layer of the visualization system being validated.

6.1 Evaluation objectives

6.1.1 FEDS-based evaluation framing (why / what / how / when)

Following FEDS, the evaluation objectives are categorized into four levels:

- Why: To prove the dashboard's effectiveness as a territorial energy decision support system (DSS) for exploration, comparison, and scenario analysis.

- What: (a) accuracy of indicators and data aggregation, (b) clarity and quality of interactions, (c) robustness and semantic consistency of scenarios.

- How: Using triangulation through analytical validation, inspection-based HCI evaluation, and scenario sensitivity tests, supported by visualization evaluation scenarios.

- When: Mainly summative (thesis evidence), with formative aspects documented via issue logs and subsequent fixes.

6.1.2 Nested-model evaluation alignment

To avoid shallow evaluation (e.g., "users liked it"), the chapter explicitly evaluates each nested layer:

Table 6.1 — Evaluation plan mapped to Munzner's Nested Model

Nested model layer	Evaluation question	Method(s)	Evidence produced
Domain problem/tasks	Does the dashboard support real planning questions?	Task design walkthrough with domain tasks	Task set completion evidence
Data/operations	Are aggregations and indicators semantically correct?	Reconciliation invariants	Validation log deviation metrics

		cross-view checks	
Encoding/interaction	Do maps/charts support valid comparisons with low cognitive friction?	Heuristic evaluation & cognitive walkthrough	Issue log Severity fixes
Algorithms/ implementation	Do computations behave consistently under perturbations?	Scenario sensitivity & monotonicity tests	Sensitivity table plots

This structure prevents “evaluation drift” (testing what is easy rather than what is scientifically necessary).

6.1.3 What “success” means for this artifact (operational criteria)

A set of measurable conditions defines success:

Analytical validity: aggregation identities hold within acceptable limits; cross-view values are consistent.

Interpretation validity: legends, units, scales, and category decompositions enable accurate reading and comparison.

Scenario validity: baseline invariance is maintained; scenario effects increase or decrease monotonically with parameter changes; effects combine coherently across different scales.

6.2 Analytical validation of indicators

This section confirms the accuracy of the dashboard’s indicator calculations and aggregation methods as analytical assertions. The main risk in multi-scale dashboards is that users draw policy conclusions from numbers that might be inconsistent across (a) scale, (b) time aggregation, or (c) component breakdown. Therefore, validation emphasizes reconciliation, invariants, and cross-view consistency, which are more robust than just “spot-checking values.”

6.2.1 Reconciliation tests (multi-scale accounting identities)

Let $\boldsymbol{x_{u,t}}$be an indicator for the unit $\boldsymbol{u}$at time $\boldsymbol{t}$. For any province $\boldsymbol{p}$with municipalities $\boldsymbol{m} \in \boldsymbol{p}$:

$$\boldsymbol{x_{p,t}} \approx \sum_{\boldsymbol{m} \in \boldsymbol{p}} \boldsymbol{x_{m,t}}$$

Similarly, for any region $\boldsymbol{r}$with provinces $\boldsymbol{p} \in \boldsymbol{r}$:

$$\boldsymbol{x_{r,t}} \approx \sum_{\boldsymbol{p} \in \boldsymbol{r}} \boldsymbol{x_{p,t}}$$

Tolerance is required only to account for missing data coverage or rounding; the evaluation must explicitly report this tolerance and the maximum deviation.

Table 6.2 — Territorial reconciliation results (reportable template)

Indicator	**Scale**	**Identity tested**	**Tolerance**	**Max deviation**	**Pass rate**
Total consumption	Province	prov = Σ mun	…	…	…
Total production	Region	reg = Σ prov	…	…	…

6.2.2 Cross-view equivalence tests (map ↔ charts ↔ exports)

A dashboard fails analytically if the same territory/time selection yields different results across views. Therefore, for each representative interaction state $\boldsymbol{s}$:

$$\boldsymbol{x}^{map}(\boldsymbol{s}) = \boldsymbol{x}^{chart}(\boldsymbol{s}) = \boldsymbol{x}^{report}(\boldsymbol{s})$$

6.2.3 Validity risks and mitigation

This evaluation explicitly checks for three failure modes (often overlooked in dashboard analyses):

- Hidden missingness (values appear complete, but coverage varies by territory).

- Unit drift (inconsistencies in GWh/MWh across different views).

- Category non-closure (sector/technology components do not sum to totals).

These are considered threats to validity, not implementation bugs, and are documented in the results.

6.3 Inspection-based HCI evaluation

A thorough evaluation should consider not just "ease of use" but also whether the interface facilitates analysis and reasoning. The literature on visual analytics evaluation clearly acknowledges that insight-focused systems require multiple evaluation approaches, including inspection, task-based walkthroughs, and assessment communication.

6.3.1 Why inspection

Inspection-based methods are used because the thesis evaluates a specialized professional tool, making large-scale user experiments often impractical within the thesis timeframe. Inspection provides systematic and reproducible evidence of interface quality through explicit tasks, severity scoring, and traceable fixes.

The evaluation includes:

- Heuristic evaluation (to identify overall usability issues),

- Cognitive walkthrough (to verify that task completion is discoverable without prior training).

6.3.2 Task set grounded in territorial planning cognition

Tasks are defined to test the reasoning loop implied by your system (spatial → temporal → comparative → scenario → communication):

Table 6.3 — Core decision tasks used in the cognitive walkthrough

Task ID	User goal	Required interface actions	Expected cognitive outcome
T1	Find a territory	Search/select on the map	Establish spatial focus
T2	Compare across scales	Switch region/province/municipality	Understand the aggregation effect
T3	Explain the temporal pattern	Switch monthly; read chart	Identify seasonality/trends
T4	Attribute composition	Interpret stacked bars/lines	Explain sector/tech contributions
T5	Communicate findings	Export report	Produce shareable evidence

6.3.3 interpretability validation, not just usability

A decision-support interface must be evaluated for interpretation correctness: can a planner correctly infer "what is dominant," "what is seasonal," and "what changed under scenario." This corresponds directly to Lam et al.'s scenario of evaluating "visual data analysis and reasoning," not merely user satisfaction.

Therefore, the inspection log includes interpretation risks such as:

ambiguous legend bins,

unclear unit/timeframe meaning ("Annual" relative to what),

insufficient explanation of stacked composition semantics.

Table 6.4 — HCI + interpretability issue log (severity-scored)

ID	Type	Observation	Risk for decision-making	Severity	Evidence (Fig.)	Mitigation
I1	Interpretability	Annual meaning is not explicit	Misread temporal scope	…	6.2	Tooltip + subtitle
I2	Visual encoding	Legend bins dense	Misjudge magnitude differences	…	6.3	Simplify bins/quantiles.
I3	Coordination	Selection state subtle	Wrong territory context	…	6.2	Stronger highlight + header

6.4 Scenario consistency and sensitivity checks

Scenario evaluation is considered a form of model-behavior validation, even when scenarios rely on indicators instead of comprehensive energy-system simulations. The aim is to demonstrate that scenarios function as logical developments of the baseline assumptions, rather than random alternative datasets.

6.4.1 Structural consistency (invariants)

The evaluation verifies invariants that must hold for any scenario engine:

- Baseline invariance: scenario parameters equal baseline ⇒ outputs equal baseline.

- Closure: totals equal the sum of components (where decomposition is shown).

- Scale coherence: region scenario effect equals the sum of provincial effects.

Table 6.5 — Scenario invariants and consistency checks

Invariant	Test design	Expected result
Baseline invariance	set scenario=params(baseline)	$\Delta = 0$
Component closure	total = Σ components	deviation ≤ tolerance
Cross-scale coherence	Δregion = Σ Δprovince	deviation ≤ tolerance

6.4.2 Sensitivity (behavior under perturbation)

Sensitivity tests evaluate whether outputs respond in the correct direction when parameters change. This is a minimal but strong behavioral validation step.

For a production multiplier $\boldsymbol{k > 1}$:

$$\boldsymbol{Prod_{scenario}(k) \geq Prod_{baseline}}$$

For a consumption reduction $\boldsymbol{k < 1}$:

$$\boldsymbol{Cons_{scenario}(k) \leq Cons_{baseline}}$$

Table 6.6 — Scenario sensitivity results (reportable format)

Parameter changed	Change	Expected direction	Observed	Pass/Fail

Solar production	+10%	total production ↑	…	…
Residential consumption	−10%	total consumption ↓	…	…

6.4.3 The semantic validity of scenarios

A common weakness in dashboard theses is treating scenarios as "UI toggles." Here, you clearly define scenario semantics: what parameters mean, which indicators must change, and which must stay the same. This shifts scenario evaluation from UI testing to decision-support validation.

6.5 Discussion of results

6.5.1 Benchmark comparison (tool class vs tool class)

To avoid superficial claims of being superior, the benchmark comparison differentiates between two types of tools: energy system planning models like EnergyPLAN and LEAP, which focus on modeling depth and simulation capabilities, often rated highly for detailed scenario analysis; and territorial planning dashboards or GIS platforms, which prioritize spatial exploration, stakeholder communication, and usability, although they may lack rigorous scenario semantics. Urban energy planning evaluations demonstrate that even the leading tools optimize different goals—whether it's depth, usability, or territorial exploration. Many GIS-supported platforms exist because model-centric tools often do not offer stakeholder-friendly exploratory interfaces.

6.5.2 Results positioning of your dashboard

Based on the evaluation structure above, the dashboard's most robust validated contributions are:

▪ Multi-scale comparability (validated through reconciliation and cross-view equivalence).

▪ Reasoning support (validated through interpretation-focused inspection, not just usability).

▪ Scenario coherence at the indicator level (validated through invariants and monotonic sensitivity).

The dashboard's limitation compared to EnergyPLAN/LEAP-class tools is not about "quality," but scope: it emphasizes territorial planning indicators and interactive sense-making rather than full sector-coupled simulation at hourly resolution. This limitation is normal and justified when clearly stated as a design boundary aligned with the planning context.

Table 6.7 — Benchmark comparison aligned to evaluation evidence

Capability	Model-centric tools (EnergyPLAN/LEAP class)	dashboard	Evidence type in this thesis
Modeling depth (sector coupling)	High	Medium	Scope statement + scenario semantics
Territorial exploration	Low–Medium	High	Fig. 6.2–6.4 + task walkthrough

Multi-scale aggregation validity	Medium	High	Tables 6.2 + 6.5
Stakeholder communication	Medium–High	High	Fig. 6.5 (report)
Interactive reasoning loop	Medium	High	Table 6.3 + issue log

6.5.3 Threats to validity and mitigation

A comprehensive evaluation chapter must address validity threats:

- Construct validity: Does "success" truly reflect the ability to plan? This is mitigated by aligning domain tasks and using nested model mapping.

- Internal validity: Are the results caused by the system rather than data anomalies? This is mitigated through reconciliation processes and invariants.

- External validity: Can these findings be applied beyond Torino? This is addressed with multi-scale testing across various regions and reporting deviations.

- Conclusion validity: Are the claims accurately supported? This is managed by framing tool-class benchmarks, distinguishing between dashboard and simulation.

Chapter 7 — Discussion, Conclusions, and Future Research

(This chapter consolidates interpretation, contributions, policy relevance, and forward-looking reflections into a single, coherent closing argument.)

7.1 Interpretation of Findings

This research focused on designing and testing a multi-scale decision-support dashboard for territorial energy planning aligned with SECAP within local governance. The findings reveal that its main value lies not in predictive accuracy but in its capacity to organize, mediate, and interpret complex territorial energy data across different scales.

The results demonstrate that the dashboard effectively facilitates core functions from the Design Science Research (DSR) framework, including sensemaking, option comparison, and scenario analysis. Clearly separating baseline data, predefined scenarios, and user configurations was crucial to avoid confusion and information overload, confirming that public-sector decision-support tools perform best when they are transparent and traceable rather than fully automated (March & Smith, 1995; Arnott & Pervan, 2012).

Analytical validation indicates that aggregated energy metrics such as self-consumption, self-sufficiency, and source-specific shares retain their meaning across various territorial levels when supported by a robust data model and logical structure. This highlights that meticulous data organization is vital for adequate visualization, beyond just being an implementation detail.

7.2 Implications for Territorial Energy Planning

From a territorial energy planning standpoint, these results reveal a ongoing disconnect between the formal requirements of planning tools, such as SECAPs, and the actual analytical methods used by local authorities. Although SECAP guidelines highlight activities like monitoring, comparing scenarios, and iterative revisions, these are often carried out with static spreadsheets and separate tools. The new dashboard bridges this gap by enabling planners to interactively examine energy trends across municipalities, provinces, and regions while remaining aligned with SECAP indicator logic. Crucially, it does not dictate decisions but instead supports informed discussions, aligning with the governance-focused approach of territorial energy planning. This observation supports existing critiques in energy planning literature, which suggest that decision-support tools should act as boundary objects—fostering dialogue between technical experts, policymakers, and stakeholders—rather than as opaque optimization systems (Star & Griesemer, 1989; Rittel & Webber, 1973).

7.3 Contributions to Decision-Support System Research

This thesis advances DSS research in three ways. First, it empirically shows how Design Science Research applies to exploratory, non-optimizing decisions, expanding beyond traditional focus on well-structured problems (Hevner et al., 2004; Peffers et al., 2007). Second, it highlights that scenario metadata and parameter transparency are core design elements, supporting interpretability, reproducibility, and trust, as emphasized in DSS research (Gregor & Hevner, 2013). Third, it argues that multi-scale aggregation functions as a decision-support tool, shaping how energy issues are framed across municipal, provincial, and regional levels.

7.4 Contributions to HCI and Visual Analytics

From an HCI and visual analytics perspective, this research sheds light on how interactive dashboards can facilitate complex decision-making in the public sector. The inspection-based HCI evaluation emphasizes the value of progressive disclosure, consistent visual encoding, and interaction features that match users' mental models of territorial governance. These findings align with previous research indicating that visual analytics systems are most effective when they ease cognitive load through clear interaction hierarchies and predictable visual responses (Shneiderman, 1996; Munzner, 2014). Additionally, the system demonstrates how visual analytics can reconcile analytical flexibility with usability—a common challenge in the literature. Instead of offering numerous controls, the dashboard limits interactions to meaningful analytical dimensions, enabling focused exploration without overwhelming users.

7.5 Policy Relevance and Institutional Implications

The proposed system at the policy level aligns with European energy governance goal,s emphasizing evidence-based planning, transparency, and local capacity enhancement. By integrating SECAP indicators into an interactive platform, the dashboard helps municipalities shift from merely fulfilling reporting requirements to ongoing analytical involvement. Institutionally, it promotes a transition from document-focused planning methods to infrastructure-supported decision-making, ensuring data, indicators, and scenarios are accessible throughout the policy process. This change impacts how local administrations assign analytical responsibilities and sustain institutional memory across multiple planning cycles.

7.6 Summary of Contributions

In summary, this research delivers:

- ✓ A DSR-compliant decision-support artifact tailored to territorial energy planning.
- ✓ A structured data and scenario model enabling consistent multi-scale analysis.
- ✓ Empirical insights into the evaluation of exploratory decision-support dashboards.
- ✓ Design principles applicable to future public-sector visual analytics systems.

7.7 Methodological Reflections and Limitations

Methodologically, the study affirms that Design Science Research is appropriate for tackling ill-structured planning challenges, while highlighting its reliance on meticulous evaluation design. The lack of extensive user studies restricts conclusions about long-term adoption and organizational effects. Nonetheless, this constraint aligns with the exploratory focus of the research and does not compromise the validity of the design insights.

7.8 Directions for Future Research

Future work could expand this research by:

▪ Conducting longitudinal studies with municipal planners to evaluate real-world decision impacts.

▪ Adding participatory features to foster collaborative scenario building.

▪ Enhancing the analytical model to include uncertainty representation and policy trade-offs.

▪ Comparing the dashboard's effectiveness with existing commercial or institutional planning tools.

7.9 Final Remarks

This thesis shows that effective decision support for territorial energy planning comes not from automation, but from well-designed analytical environments that acknowledge the complexity, uncertainty, and institutional context of public decision-making. By integrating Design Science Research, SECAP domain expertise, and human-centered visual analytics, it lays a foundation for future research and practice at the crossroads of energy planning and digital decision-support tools.

References:

European Commission, Joint Research Centre & Covenant of Mayors. (2018). Guidebook: How to develop a **Sustainable Energy and Climate Action Plan (SECAP)**. Publications Office of the European Union. ISBN 978-92-79-96927-0. https://op.europa.eu/en/publication-detail/-/publication/fd75e1e2-f132-11e8-9982-01aa75ed71a1

European Commission, Directorate-General for Communication. (2024). Italy – Final updated **National Energy and Climate Plan 2021–2030 (NECP)**. European Commission. https://commission.europa.eu/publications/italy-final-updated-necp-2021-2030-submitted-2024_en

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). *Design science in information systems research*. **MIS Quarterly, 28**(1), 75–105. https://doi.org/10.2307/25148625

March, S. T., & Smith, G. F. (1995). *Design and natural science research on information technology*. **Decision Support Systems, 15**(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). *A design science research methodology for information systems research*. **Journal of Management Information Systems, 24**(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Norman, D. A. (2013). *The design of everyday things* (Revised and expanded edition). Basic Books.

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). *Designing the user interface: Strategies for effective human–computer interaction* (6th ed.). Pearson.

Simon, H. A. (1955). A behavioral model of rational choice. *The Quarterly Journal of Economics, 69*(1), 99–118. https://doi.org/10.2307/1884852

Ware, C. (2013). *Information visualization: Perception for design* (3rd ed.). Morgan Kaufmann.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human–computer interaction*. Lawrence Erlbaum Associates.

Delmastro, C., Mutani, G., & Schranz, L. (2020). The role of spatial planning in the energy transition: A review of methods and tools. *Renewable and Sustainable Energy Reviews, 133*, 110242. https://doi.org/10.1016/j.rser.2020.110242

Lowitzsch, J., Hoicka, C. E., & van Tulder, F. J. (2020). Renewable energy communities under the 2019 European Clean Energy Package. *Energy Policy, 138*, 111212. https://doi.org/10.1016/j.enpol.2019.111212

Pfenninger, S., Hawkes, A., & Keirstead, J. (2014).
Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews, 33*, 74–86.
https://doi.org/10.1016/j.rser.2014.02.003

Roberts, J., Frieden, D., & d'Herbemont, S. (2019).
Energy community definitions, barriers and drivers. *Energy Policy, 137*, 111106.
https://doi.org/10.1016/j.enpol.2019.111106

Trutnevyte, E., et al. (2019).
Societal transformations in models for energy and climate policy. *Nature Climate Change, 9*(10), 754–760.
https://doi.org/10.1038/s41558-019-0577-z

van der Heijden, K. (2005).
Scenarios: The art of strategic conversation (2nd ed.). Wiley.

Arnott, D., & Pervan, G. (2005).
A critical analysis of decision support systems research. *Journal of Information Technology, 20*(2), 67–87.
https://doi.org/10.1057/palgrave.jit.2000035

Geertman, S., & Stillwell, J. (2009).
Planning support systems: Content, issues and trends. In S. Geertman & J. Stillwell (Eds.), *Planning support systems best practice and new methods* (pp. 1–26). Springer. https://doi.org/10.1007/978-1-4020-8952-7_1

Ferrari, S. (2019). *Assessment of tools for urban energy planning*.

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., & Carpendale, S. (2012). Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*.

Munzner, T. (2009). *A nested model for visualization design and validation*.

Roberts, J. C. (2007). Coordinated multiple views for exploratory visualization. *Proceedings of Coordinated and Multiple Views in Exploratory Visualization*.

Appendix

Hooks- useMonthlyData.ts

```
// src/hooks/useMonthlyData.ts
import { useEffect, useMemo, useState } from "react";

const API_BASE = "http://localhost:5000";

// Backend uses "comune"
export type BackendLevel = "region" | "province" | "comune";

export type SeriesPoint = { x: number; value_mwh: number };

export type CombinedChartPoint = {
  month: string; // "Jan".."Dec"

  // totals (GWh)
  totalConsumption?: number;
  totalProduction?: number;

  // production breakdown (stacked bars) (GWh)
  solar?: number;
  wind?: number;
  hydroelectric?: number;
  geothermal?: number;
  biomass?: number;

  // consumption breakdown (lines) (GWh)
  residential?: number;
  primary?: number;
  secondary?: number;
  tertiary?: number;
};

const MONTH_LABELS = [
  "Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec",
];

function mwhToGwh(mwh: number): number {
  return mwh / 1000;
}

function buildSeriesUrl(params: Record<string, string | number | undefined>) {
  const usp = new URLSearchParams();
  Object.entries(params).forEach(([k, v]) => {
    if (v === undefined || v === null || v === "") return;
    usp.set(k, String(v));
  });
  return `${API_BASE}/charts/series?${usp.toString()}`;
}
```

```
async function fetchSeries(url: string, signal: AbortSignal): Promise<SeriesPoint[]> {
  const res = await fetch(url, { signal });
  if (!res.ok) {
    const text = await res.text().catch(() => "");
    throw new Error(
      `Failed: ${res.status} ${res.statusText}${text ? ` — ${text}` : ""}`
    );
  }
  const json = (await res.json()) as SeriesPoint[];
  return Array.isArray(json) ? json : [];
}

//  backend param names confirmed by you
function codeParamKey(level: BackendLevel): "region_code" | "province_code" | "comune_code" {
  if (level === "region") return "region_code";
  if (level === "province") return "province_code";
  return "comune_code";
}

export default function useMonthlyData(
  level: BackendLevel,
  territoryCode: number | null | undefined,
  year: number,
  scenario = 0
) {
  const [data, setData] = useState<CombinedChartPoint[]>([]);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState<string | null>(null);

  const hasTerritory = !!territoryCode;

  const empty = useMemo<CombinedChartPoint[]>(
    () =>
      MONTH_LABELS.map((m) => ({
        month: m,
        totalConsumption: 0,
        totalProduction: 0,
        solar: 0,
        wind: 0,
        hydroelectric: 0,
        geothermal: 0,
        biomass: 0,
        residential: 0,
        primary: 0,
        secondary: 0,
        tertiary: 0,
      })),
    []
  );

  useEffect(() => {
    if (!territoryCode) {
      setData([]);
      setError(null);
      setLoading(false);
      return;
```

```
  }

  const controller = new AbortController();

  async function run() {
    try {
      setLoading(true);
      setError(null);

      const key = codeParamKey(level);

      const baseParams: Record<string, string | number | undefined> = {
        level,
        resolution: "monthly",
        year,
        scenario,
        [key]: territoryCode,
      };

      // totals
      const totalConsUrl = buildSeriesUrl({
        ...baseParams,
        domain: "consumption",
      });

      const totalProdUrl = buildSeriesUrl({
        ...baseParams,
        domain: "production",
      });

      // consumption categories
      const consResidentialUrl = buildSeriesUrl({
        ...baseParams,
        domain: "consumption",
        category_code: "cons_domestic",
      });

      const consPrimaryUrl = buildSeriesUrl({
        ...baseParams,
        domain: "consumption",
        category_code: "cons_primary",
      });

      const consSecondaryUrl = buildSeriesUrl({
        ...baseParams,
        domain: "consumption",
        category_code: "cons_secondary",
      });

      const consTertiaryUrl = buildSeriesUrl({
        ...baseParams,
        domain: "consumption",
        category_code: "cons_tertiary",
      });

      // production base_groups
      const prodSolarUrl = buildSeriesUrl({
        ...baseParams,
```

```
    domain: "production",
    base_group: "solar",
  });

  const prodWindUrl = buildSeriesUrl({
    ...baseParams,
    domain: "production",
    base_group: "wind",
  });

  const prodHydroUrl = buildSeriesUrl({
    ...baseParams,
    domain: "production",
    base_group: "hydroelectric",
  });

  const prodGeoUrl = buildSeriesUrl({
    ...baseParams,
    domain: "production",
    base_group: "geothermal",
  });

  const prodBiomassUrl = buildSeriesUrl({
    ...baseParams,
    domain: "production",
    base_group: "biomass",
  });

  const [
    totalCons,
    totalProd,
    consResidential,
    consPrimary,
    Secondary,
    consTertiary,
    prodSolar,
    prodWind,
    prodHydro,
    prodGeo,
    prodBiomass,
  ] = await Promise.all([
    fetchSeries(totalConsUrl, controller.signal),
    fetchSeries(totalProdUrl, controller.signal),
    fetchSeries(consResidentialUrl, controller.signal),
    fetchSeries(consPrimaryUrl, controller.signal),
    fetchSeries(consSecondaryUrl, controller.signal),
    fetchSeries(consTertiaryUrl, controller.signal),
    fetchSeries(prodSolarUrl, controller.signal),
    fetchSeries(prodWindUrl, controller.signal),
    fetchSeries(prodHydroUrl, controller.signal),
    fetchSeries(prodGeoUrl, controller.signal),
    fetchSeries(prodBiomassUrl, controller.signal),
  ]);

  const byMonth = new Map<number, CombinedChartPoint>();
  for (let i = 1; i <= 12; i++) byMonth.set(i, { ...empty[i - 1] });
```

```
        const apply = (series: SeriesPoint[], key: keyof CombinedChartPoint) =>
{
          for (const p of series) {
            const m = p.x; // 1..12
            const row = byMonth.get(m);
            if (!row) continue;
            (row[key] as number) = mwhToGwh(p.value_mwh ?? 0);
          }
        };

        apply(totalCons, "totalConsumption");
        apply(totalProd, "totalProduction");

        apply(consResidential, "residential");
        apply(consPrimary, "primary");
        apply(consSecondary, "secondary");
        apply(consTertiary, "tertiary");

        apply(prodSolar, "solar");
        apply(prodWind, "wind");
        apply(prodHydro, "hydroelectric");
        apply(prodGeo, "geothermal");
        apply(prodBiomass, "biomass");

        setData(Array.from(byMonth.values()));
      } catch (e: any) {
        if (e?.name === "AbortError") return;
        setError(e?.message || "Failed to load monthly series");
        setData([]);
      } finally {
        setLoading(false);
      }
    }

    run();
    return () => controller.abort();
  }, [level, territoryCode, year, scenario, empty]);

  return { data, loading, error, hasTerritory };
}
```

Hooks- useGeoData.ts

```
// src/components/maps/hooks/useGeoData.ts
import { useEffect, useMemo, useRef, useState } from "react";
import type { FeatureCollection, Geometry } from "geojson";

type BackendLevel = "region" | "province" | "comune";
type AnyFC = FeatureCollection<Geometry, any>;

type ValuesRow = {
  cod_reg?: number | string | null;
  reg_cod?: number | string | null;
```

```
  cod_prov?: number | string | null;
  prov_cod?: number | string | null;
  pro_com?: number | string | null;
  mun_cod?: number | string | null;
  cod_com?: number | string | null;
  value_mwh: number | string | null;
};

const GEO_API = "http://localhost:5000/map/territories";
const VALUES_API = "http://127.0.0.1:5000/charts/values";

function toNum(x: unknown): number | null {
  if (typeof x === "number" && Number.isFinite(x)) return x;
  if (typeof x === "string") {
    const n = Number(x.trim());
    return Number.isFinite(n) ? n : null;
  }
  return null;
}

function normalizeLevel(level: unknown): BackendLevel | null {
  if (level === "region" || level === "province" || level === "comune") return
level;

  // common frontend label → backend label
  if (level === "municipality") return "comune";

  // allow string-ish values
  if (typeof level === "string") {
    const v = level.trim().toLowerCase();
    if (v === "region") return "region";
    if (v === "province") return "province";
    if (v === "comune") return "comune";
    if (v === "municipality") return "comune";
  }

  return null;
}

function simplifyFor(level: BackendLevel) {
  if (level === "region") return 0.02;
  if (level === "province") return 0.005;
  return 0.01;
}

function codeFromValuesRow(level: BackendLevel, r: ValuesRow): number | null {
  if (level === "region") return toNum((r as any).cod_reg ?? (r as any).reg_cod);
  if (level === "province") return toNum((r as any).cod_prov ?? (r as
any).prov_cod);
  return toNum((r as any).pro_com ?? (r as any).mun_cod ?? (r as any).cod_com);
}

function valueFromRow(r: ValuesRow): number | null {
  if (typeof r.value_mwh === "number") return Number.isFinite(r.value_mwh) ?
r.value_mwh : null;
  return toNum(r.value_mwh);
}
```

```
export type UseGeoDataArgs = {
  //  changed: tolerate runtime undefined / UI labels
  level: BackendLevel | "municipality" | string | null | undefined;

  domain: string;
  resolution: "annual" | "monthly" | "hourly";
  year: number;
  scenario: number;
  baseGroup?: "domestic" | "primary" | "secondary" | "tertiary";
};

export type UseGeoDataResult = {
  geo: AnyFC | null;
  valuesMap: Map<number, number>;

  phase: "idle" | "geo_loading" | "geo_ready" | "values_loading" | "ready" |
"error";

  loadingGeo: boolean;
  loadingValues: boolean;
  error: string | null;

  debug: {
    geoUrl: string;
    valuesUrl: string;
    geoFeatures: number;
    valuesMapped: number;
    geoSampleProps: any[];
    valuesSamplePairs: Array<{ code: number; value: number }>;
  };
};

export function useGeoData({
  level,
  domain,
  resolution,
  year,
  scenario,
}: UseGeoDataArgs): UseGeoDataResult {
  const normalizedLevel = useMemo(() => normalizeLevel(level), [level]);
  const enabled = normalizedLevel != null;

  const [geo, setGeo] = useState<AnyFC | null>(null);
  const [valuesMap, setValuesMap] = useState<Map<number, number>>(new Map());

  const [loadingGeo, setLoadingGeo] = useState(false);
  const [loadingValues, setLoadingValues] = useState(false);
  const [error, setError] = useState<string | null>(null);

  const [phase, setPhase] = useState<UseGeoDataResult["phase"]>("idle");

  // monotonically increasing request id to ignore stale responses
  const reqIdRef = useRef(0);

  const geoUrl = useMemo(() => {
    if (!normalizedLevel) return "";
    const qs = new URLSearchParams({
      level: normalizedLevel,
```

```
    simplify: String(simplifyFor(normalizedLevel)),
  });
  return `${GEO_API}?${qs.toString()}`;
}, [normalizedLevel]);

const valuesUrl = useMemo(() => {
  if (!normalizedLevel) return "";
  const qs = new URLSearchParams({
    level: normalizedLevel,
    resolution,
    year: String(year),
    domain,
    scenario: String(scenario),
  });
  return `${VALUES_API}?${qs.toString()}`;
}, [normalizedLevel, resolution, year, domain, scenario]);

//  Ordered pipeline: geo -> render -> values -> recolor
useEffect(() => {
  //  HARD GUARD: never fetch with invalid level
  if (!enabled) {
    // Do not nuke existing data; just go idle.
    setLoadingGeo(false);
    setLoadingValues(false);
    setPhase("idle");
    setError(null);

    console.warn("[PIPE] skip: invalid level", { level, normalizedLevel });
    return;
  }

  const controller = new AbortController();
  const myReqId = ++reqIdRef.current;

  (async () => {
    // reset for a clean run (only when enabled)
    setError(null);
    setGeo(null);
    setValuesMap(new Map());

    setLoadingGeo(true);
    setLoadingValues(false);
    setPhase("geo_loading");

    try {
      console.log("[PIPE] geo:", geoUrl);
      const resGeo = await fetch(geoUrl, { signal: controller.signal });
      if (!resGeo.ok) throw new Error(`Geo load failed: ${resGeo.status}`);

      const fc = (await resGeo.json()) as AnyFC;

      // ignore stale responses
      if (reqIdRef.current !== myReqId) return;

      setGeo(fc);
      setLoadingGeo(false);
      setPhase("geo_ready");
```

```
      // Now fetch values (after geo is visible)
      setLoadingValues(true);
      setPhase("values_loading");

      console.log("[PIPE] values:", valuesUrl);
      const resVals = await fetch(valuesUrl, { signal: controller.signal });
      if (!resVals.ok) throw new Error(`Values load failed: ${resVals.status}`);

      const rows = (await resVals.json()) as ValuesRow[];
      const m = new Map<number, number>();

      for (const r of rows) {
        const code = codeFromValuesRow(normalizedLevel, r);
        const val = valueFromRow(r);
        if (code == null || val == null) continue;
        m.set(code, val);
      }

      if (reqIdRef.current !== myReqId) return;

      setValuesMap(m);
      setLoadingValues(false);
      setPhase("ready");
    } catch (e: any) {
      if (e?.name === "AbortError") return;
      console.error(e);

      if (reqIdRef.current !== myReqId) return;

      setError(e?.message ?? "Fetch error");
      setLoadingGeo(false);
      setLoadingValues(false);
      setPhase("error");
    }
  })();

  return () => controller.abort();
  // important: geoUrl/valuesUrl already encode normalizedLevel
}, [enabled, geoUrl, valuesUrl, level, normalizedLevel]);

const debug = useMemo(() => {
  const geoFeatures = geo?.features?.length ?? 0;
  const valuesMapped = valuesMap.size;

  const geoSampleProps = (geo?.features ?? [])
    .slice(0, 5)
    .map((f: any) => f?.properties ?? {});

  const valuesSamplePairs = Array.from(valuesMap.entries())
    .slice(0, 5)
    .map(([code, value]) => ({ code, value }));

  return {
    geoUrl,
    valuesUrl,
    geoFeatures,
    valuesMapped,
    geoSampleProps,
```

```
      valuesSamplePairs,
    };
  }, [geo, valuesMap, geoUrl, valuesUrl]);

  return {
    geo,
    valuesMap,
    phase,
    loadingGeo,
    loadingValues,
    error,
    debug,
  };
}
```