ARES: Adaptive Data Provisioning for Smart Environments

Giovanni Donato Gallo, Daniela Micucci
Department of Informatics, Systems, and Communication
University of Milano-Bicocca
Viale Sarca 336, Edificio U14, 20126 Milan, Italy
giovanni.gallo@unimib.it, daniela.micucci@unimib.it

***Abstract*—Modern smart-city and intelligent-environment systems increasingly rely on heterogeneous and dynamically evolving data sources exposing different quality characteristics over time. Existing adaptive provisioning approaches typically perform provider selection at each invocation, introducing runtime overhead and potentially unstable behavior under noisy or delayed observations.**

This work presents ARES (Adaptive Runtime Execution over heterogeneous Sources), a configuration-driven architectural approach for adaptive data provisioning. ARES treats provider selection as an explicit architectural concern by maintaining persistent runtime configurations updated according to externally defined adaptation policies. The approach separates candidate providers from the active configuration and decouples global adaptation decisions from local provider behaviors. ARES is particularly suitable for smart-city ecosystems involving IoT infrastructures, edge-cloud environments, urban monitoring platforms, smart mobility services, and digital-health applications. By explicitly modeling the temporal dimension of provider-selection decisions, ARES enables reasoning about responsiveness, stability, and adaptation overhead in dynamic environments.

***Index Terms*—Adaptive Data Provisioning, Software Architecture, Self-Adaptive Systems, QoS-Aware Systems**

I. Introduction

Modern smart-city ecosystems increasingly operate over highly heterogeneous and distributed infrastructures involving IoT devices, cloud-edge services, wearable sensors, mobile applications, and user-generated information. In these environments, the same capability or information can often be obtained through multiple alternative providers exposing different quality characteristics.

Current adaptive provisioning approaches typically perform provider selection at each invocation according to the latest observed conditions [1]. This strategy improves responsiveness, but it may also introduce instability, unnecessary runtime overhead, and excessive sensitivity to noisy or delayed observations. Recent studies have emphasized the need to minimize unnecessary re-optimization while maintaining stable placements over time [2].

This work presents ARES, a configuration-driven architecture that explicitly treats provider selection as a persistent architectural concern. Instead of continuously re-evaluating providers at runtime, ARES maintains an active configuration updated according to configurable adaptation policies.

II. ARES Architectural Approach

Figure 1 summarizes the main architectural abstractions of ARES. The architecture explicitly separates the catalog of available capabilities and providers from the active runtime configuration used during execution.

Applications interact with the system by requesting abstract capabilities rather than directly invoking concrete providers. Multiple heterogeneous providers may expose the same capability while exhibiting different quality characteristics and contextual constraints. The active runtime configuration determines which provider is currently associated with each capability.

Unlike traditional approaches based on continuous per-invocation reselection, ARES introduces the notion of persistent configuration decisions. Once a provider is selected for a capability, the decision remains active across multiple invocations until explicit reassessment policies trigger a reconfiguration process.

The architecture therefore introduces a temporal dimension in adaptive provisioning strategies. In particular, ARES explicitly models: (i) when provider-selection decisions should be re-evaluated; (ii) how long a configuration should remain valid; and (iii) how adaptation policies affect stability and runtime overhead. This perspective enables adaptive provisioning strategies that balance responsiveness and stability according to application-specific requirements.

ARES adopts a capability-driven abstraction model in which applications specify what information is required independently from how such information is obtained. Capabilities therefore represent abstract functionalities that can be realized through multiple heterogeneous providers. This abstraction decouples application logic from provider-specific implementation details and enables flexible adaptation strategies.

For each capability, the architecture maintains a set of candidate providers associated with quality-related information such as reliability, latency, freshness, or contextual constraints. Runtime requests are resolved through the currently active configuration without continuously repeating provider-selection operations.

ARES also distinguishes between global adaptation decisions and local provider behaviors [3]. Global adaptation mechanisms determine when reassessment and reconfiguration

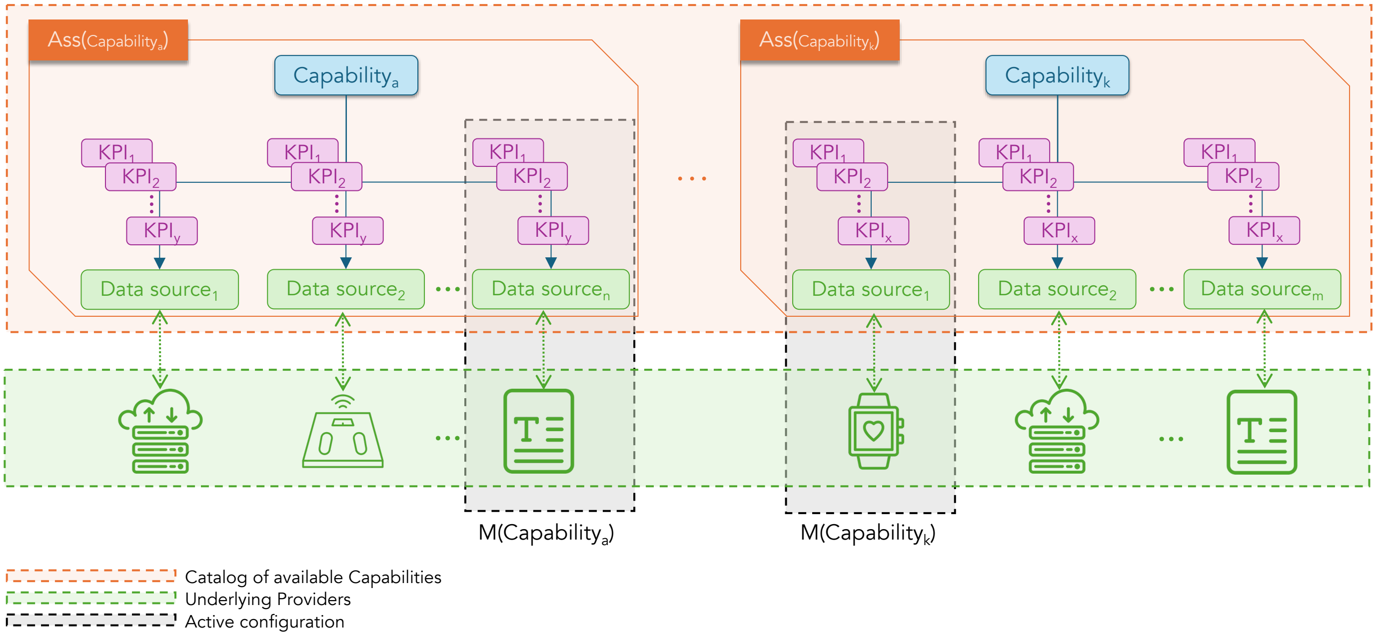

Fig. 1. Conceptual overview of the ARES architectural model.

should occur, while local providers independently manage data acquisition, caching strategies, quality estimation, or contextual adaptation policies.

This separation supports different adaptation granularities and enables heterogeneous realization mechanisms across cloud, edge, and IoT infrastructures. Moreover, ARES can be combined with existing self-adaptive systems and digital-twin-based infrastructures, where providers may expose dynamically evolving quality characteristics, enabling flexible realization strategies across cloud, edge, and IoT environments.

Finally, ARES enables explicit reasoning about decision frequency, configuration stability, adaptation overhead, and quality trade-offs. By explicitly managing configuration persistence, it reduces excessive adaptation frequency and mitigates instability due to noisy or delayed observations, while still preserving adaptive capabilities under changing operating conditions.

III. Application Scenarios in Smart Cities

ARES is particularly suitable for smart-city ecosystems involving heterogeneous IoT infrastructures, edge-cloud environments, and dynamic contextual conditions [4].

In smart mobility systems, routing and navigation services may dynamically rely on alternative providers exposing traffic, congestion, transportation, or contextual information with different quality guarantees. ARES enables stable and policy-driven provider selection while limiting unnecessary runtime reconfiguration.

In urban monitoring platforms, heterogeneous sensing infrastructures may provide environmental observations such as pollution, weather, noise, or crowding levels. ARES supports adaptive provisioning strategies capable of balancing responsiveness and configuration stability.

The approach is also relevant for digital-health and assisted-living ecosystems integrating wearable sensors, cloud services, and manual user inputs under dynamic operating conditions.

IV. Conclusion

This extended abstract presented ARES, a configuration-driven architectural approach for adaptive data provisioning in heterogeneous and dynamic environments. The approach explicitly models provider-selection persistence and adaptation timing as architectural concerns.

ARES enables reasoning about the trade-off between responsiveness, stability, and adaptation overhead, while supporting integration with smart-city, IoT, digital-health, and intelligent-environment ecosystems.

Future work will explore richer adaptation policies, coordination across multiple capabilities and providers, and architectural integration within large-scale smart-city and intelligent-environment platforms.

References

[1] J. L. Herrera, J. Berrocal, S. Forti *et al.*, "Continuous qos-aware adaptation of cloud-iot application placements," *Computing*, vol. 105, pp. 2037–2059, 2023. [Online]. Available: https://doi.org/10.1007/s00607-023-01153-1

[2] G. R. Russo, D. Ferrarelli, D. Pasquali, V. Cardellini, and F. L. Presti, "Qos-aware offloading policies for serverless functions in the cloud-to-edge continuum," *Future Generation Computer Systems*, vol. 156, pp. 1–15, 2024.

[3] T. Wong, M. Wagner, and C. Treude, "Self-adaptive systems: A systematic literature review across categories and domains," *Information and Software Technology*, vol. 148, p. 106934, 2022.

[4] A. Hakiri, A. Gokhale, S. B. Yahia, and N. Mellouli, "A comprehensive survey on digital twin for future networks and emerging internet of things industry," *Computer Networks*, vol. 244, p. 110350, 2024.