Journal of Intelligent Systems and Future Computing

ISSN 3117 - 4698

https://jisfc.com/index.php/jisfc/index

Semantic Web Technologies for Interoperable Smart Environment Integration

Gurumurthy V[1]*, Padmanabhan D[2] and Parthasarathy S[3]

[1]Department of Computer Science and Engineering, Amaravathi Institute of Cyber-Physical Systems, Singapore Email: vijay.gurumurthy@amaravathi.edu

ORCID: https://orcid.org/3184-8815-1415-5180

[2]Department of Robotics and Automation, Sharavati Center for Robotics and Automation, South Korea Email: divya.padmanabhan@sharavati.edu

ORCID: https://orcid.org/5435-6491-6425-1301

[3]Department of Computational Intelligence, Amaravathi Institute of Cyber-Physical Systems, Philippines Email: sundar.parthasarathy@amaravathi.edu

ORCID: https://orcid.org/8992-6233-1262-2285

ABSTRACT

Smart environments are assembled from devices made by many independent vendors, and getting those devices to agree on what their data means remains a harder problem than moving the bytes between them. This article synthesizes the literature on applying semantic web technologies -- RDF and OWL as a formal substrate, purpose-built domain ontologies, and the discovery and matching infrastructure needed to reconcile vocabularies never designed to align -- to interoperable smart environment integration. We trace the technique from foundational ontology-engineering principles and the original Semantic Web vision through the W3C/OGC Semantic Sensor Network family and the industry-consensus SAREF ontology for smart appliances, to lightweight variants built for constrained devices, cross-domain reasoning frameworks, and a recent smart-city middleware whose load-tested numbers make concrete where semantic reasoning starts to strain under real concurrent demand. Read together, the evidence supports semantic web technologies as a coherent, increasingly mature pattern for smart environment interoperability, while leaving reasoning-layer scalability, persistent ontology fragmentation, and inconsistent evaluation practice as an open research agenda.

Keywords: semantic web; ontology; Internet of Things; smart environment interoperability; knowledge graphs

Citation: Gurumurthy V, Padmanabhan D and Parthasarathy S [2023]. Semantic Web Technologies for Interoperable Smart Environment Integration. **DOI:** https://doi.org/10.5281/zenodo.22843883

Copyright: © 2023 by the authors. Open access under CC BY 4.0 license.

Article Information: *Received:* 11 April 2023 *Accepted:* 10 July 2023 *Published:* 4 October 2023

Article Class: *Research Article*

1. Introduction

A modern smart environment, whether a single-family home, a hospital ward, or a district-scale smart city deployment, is rarely built by one vendor. It is assembled, piecemeal and over years, from thermostats, door locks, occupancy sensors, energy meters, wearables, and building-management controllers made by dozens of manufacturers, each speaking its own proprietary protocol and describing its own data with its own private vocabulary. The devices themselves have gotten cheap and capable — surveys of the Internet of Things landscape have long described this proliferation of heterogeneous, independently manufactured smart objects as the field's defining characteristic rather than an incidental detail [1] — and what has not gotten easier is getting them to agree on what a "temperature reading" or an "occupancy event" actually means well enough that a rule engine, a caregiver dashboard, or another vendor's device can act on it correctly. This is not primarily a networking problem — protocols like MQTT, CoAP, and plain HTTP already move bytes between devices and servers perfectly well — it is a problem of meaning, and it is precisely the problem the Semantic Web research program was built to address two decades before the Internet of Things existed as a phrase [2].

The Semantic Web's founding proposition, laid out by Berners-Lee, Hendler, and Lassila, was that machine-readable meaning, not merely machine-readable syntax, could be layered on top of the existing Web using shared, formally defined vocabularies — ontologies — that let independently built systems interpret the same data consistently [2]. Fifteen years on, Shadbolt, Berners-Lee, and Hall revisited that vision and found it only partially realized: the formal machinery (RDF, OWL, SPARQL) had matured, but the promised network effects of shared vocabularies had been slower to arrive than hoped, in part because building and agreeing on ontologies turned out to be a genuinely hard social and technical problem, not just an engineering one [3]. The Internet of Things gave this older research program a new and arguably better-suited proving ground. Where the original Semantic Web vision struggled to get millions of independent web publishers to agree on shared markup, the IoT interoperability problem is narrower and more tractable: a comparatively small number of device classes, sensor modalities, and application domains, each of which a focused standards effort can plausibly cover with a purpose-built ontology.

This article synthesizes the literature on applying semantic web technologies — ontologies, linked data, knowledge graphs, and the standards built around them — to the specific problem of making smart environments interoperable across vendors, device classes, and application domains. We are not describing a single technology so much as a layered stack: foundational knowledge-representation formalisms (RDF, OWL, and the ontology-engineering discipline built around them); domain ontologies purpose-built for sensing and actuation (the W3C/OGC Semantic Sensor Network family, and the ETSI/oneM2M Smart Appliances REFerence ontology); middleware and architectural patterns that put those ontologies to work at runtime; and the harder, less-solved problem of aligning and matching ontologies when two systems that were never designed to talk to each other need to anyway. We close with an honest account of where this stack still falls short of its promise, echoing Shadbolt et al.'s original caution that formal machinery alone does not guarantee adoption [3].

It is worth being explicit about what this article is and is not. It is a literature synthesis, not a report of new experiments; every quantitative figure we cite is drawn from a published, DOI-verifiable source, and we flag prominently where the evidence base is thin. The IoT interoperability literature itself is candid that most published systems are validated on modest testbeds or simulated deployments rather than at the scale of a real multi-vendor smart city, a point that recurs across several of the architectures reviewed below and that we return to in our discussion of open challenges.

2. Foundations of the Semantic Web

The technical case for the Semantic Web rests on a layered stack of standards, but its intellectual foundation is older and simpler than any of them: an ontology, in the sense computer science borrowed from philosophy, is a formal, explicit specification of a shared conceptualization — an agreed vocabulary of concepts and the relationships between them, precise enough that a machine can reason over it [4]. Gruber's original formulation, developed in the context of knowledge sharing between AI systems rather than the Web specifically, is still the most commonly cited definition in the semantic web literature, and it identifies exactly the property that makes ontologies useful for IoT

interoperability: a device from vendor A and a device from vendor B can interoperate not because they share code, but because they commit to the same conceptualization of what a "sensor," an "observation," or a "capability" is.

Turning that philosophical commitment into engineering practice required its own discipline. Studer, Benjamins, and Fensel's widely cited treatment of knowledge engineering principles and methods lays out the practical questions that any ontology-building effort has to answer — what level of formality is needed, how concepts should be organized into a taxonomy, how to keep an ontology consistent as it grows, and how to evaluate whether an ontology actually captures the domain it claims to model [5]. These questions are not abstract; they show up directly in the design choices the IoT ontologies discussed later in this article had to make, particularly around how much detail to standardize (a rich, deeply structured ontology captures more nuance but is harder for every vendor to adopt correctly) versus how much to leave to downstream extension.

The Resource Description Framework (RDF) supplies the syntactic substrate this conceptual work sits on: a simple, uniform way to express any statement as a subject-predicate-object triple, which turns out to be expressive enough to represent everything from a sensor's manufacturer to the physical quantity it measures to the building it sits in, all in the same uniform graph structure. The Web Ontology Language (OWL), layered on top of RDF, adds the formal semantics — class hierarchies, property restrictions, cardinality constraints — that let a reasoner infer new facts from asserted ones, for instance deriving that a device described only as a "thermostat" is also, by virtue of the class hierarchy, a "temperature sensor" and an "actuator." This inference capability is what distinguishes semantic interoperability from merely agreeing on a shared schema: a schema tells you what fields exist, an ontology with formal semantics lets a system reason about what follows from the data it already has. Bizer, Heath, and Berners-Lee's account of Linked Data adds the publication discipline that makes this graph model useful across organizational boundaries in the first place: use URIs to name things, make those URIs dereferenceable to useful RDF descriptions, and link outward to related resources described elsewhere, so that a sensor's semantic description in one system can point directly at, rather than merely resemble, the building or occupant it is associated with in another [6].

Querying this graph structure at scale is itself a solved problem in a way that is easy to take for granted: SPARQL, the standard RDF query language, lets an application ask graph-pattern questions of arbitrary complexity — "find every sensor of type temperature located in a room whose occupant has a medical condition requiring climate control" — against a store that may span data contributed by many independent publishers, without any of those publishers having agreed in advance on exactly this query. This decoupling of publication from anticipated use is, in our reading, the property that most clearly distinguishes semantic interoperability from a conventional integration approach built on fixed APIs and hand-written adapters: a fixed API answers the questions its designer anticipated, while a well-populated RDF graph with a general-purpose query language answers questions nobody anticipated, provided the underlying data was published with enough semantic fidelity to support the inference the query requires.

The practical payoff of organizing IoT and Web data this way at scale is captured well by Hogan et al.'s comprehensive survey of knowledge graphs, which treats the RDF/OWL graph model not as an academic curiosity but as the substrate underlying real, large-scale deployed systems — search engine knowledge panels, enterprise data integration, and increasingly IoT platforms — and systematically catalogs the representation choices, reasoning techniques, and open research questions that arise once a knowledge graph grows past toy scale [7]. Their survey is a useful bridge between the older, more theoretical Semantic Web literature and the newer knowledge-graph terminology that much of the recent applied IoT literature has adopted; the underlying formalism is the same RDF/OWL stack, but the framing has shifted from "the Web of documents annotated with meaning" toward "a queryable graph of entities and relationships that happens to be populated partly by sensors."

3. Ontologies for Sensors, Devices, and Smart Environments

The single most consequential piece of standards work for semantic IoT interoperability is the Semantic Sensor Network (SSN) ontology, originally developed by a W3C

Incubator Group and documented by Compton et al., which set out to provide a general-purpose vocabulary for describing sensors, the observations they produce, the properties those observations measure, and the platforms sensors are deployed on — independent of any specific sensor manufacturer, measurement domain, or deployment context [8]. The original SSN ontology was deliberately broad and, as the field gained experience deploying it, proved in places to be more heavyweight than many applications needed. A joint W3C and OGC working group subsequently revised it into a modular structure: a lightweight core pattern called SOSA (Sensor, Observation, Sample, and Actuator) that captures the minimal vocabulary most applications actually use, with the fuller SSN ontology built as an extension on top of it for applications that need the richer semantics — a redesign documented by Haller et al. as a joint standard explicitly intended to lower the barrier to adoption for lighter-weight IoT deployments while preserving compatibility with the original ontology's more elaborate modeling for applications that need it [9]. This SOSA/SSN split is, in our reading, the single most important lesson the semantic sensor network community learned from a decade of deployment experience: a maximally expressive ontology that is too heavyweight for constrained devices to actually use is not, in practice, more interoperable than no ontology at all, and the pragmatic fix was not to abandon formal rigor but to layer it, so an application only pays the complexity cost for the semantics it actually needs.

Table 1: Representative semantic web ontologies and standards for smart environment interoperability, the layer of the stack each addresses, and its defining contribution as reported in the cited source. [8,9,10,11,18,19].

Ontology / Standard	Source	Stack Layer	Defining Contribution
SSN (Semantic Sensor Network)	Compton et al. [8]	Sensing, observation, platform description	General-purpose W3C vocabulary for sensors, observations and measured properties
SOSA / modular SSN	Haller et al. [9]	Lightweight sensing core + extensible full model	Joint W3C/OGC redesign splitting a minimal core (SOSA) from the fuller SSN extension
SAREF	Daniele et al. [10]	Smart appliances and building automation	Industry-consensus ETSI ontology for appliance functions, states and commands
IoT-Lite	Bermudez-Edo et al. [11]	Constrained-device semantic description	Lightweight SSN extension reducing reasoning/storage overhead for limited hardware
LOV (Linked Open Vocabularies)	Vandenbussche et al. [18]	Vocabulary discovery and reuse	Curated, quality-checked catalog of published RDF vocabularies with reuse metadata
INTER-IoT semantic mediation	Ganzha et al. [19]	Cross-platform semantic mediation	Translation layer reconciling different IoT platforms' native data models

SSN/SOSA describes sensing in the abstract, but a smart environment also needs a vocabulary for the appliances, actuators, and building-management concepts specific to home and building automation, and this is the gap the Smart Appliances REFerence (SAREF) ontology was built to fill. Daniele, den Hartog, and Roes describe SAREF's origin as an explicitly industry-driven effort, developed under ETSI in close collaboration with appliance manufacturers rather than purely by academic ontology engineers, with the stated goal of providing a shared, minimal vocabulary that different vendors' smart appliances could use to describe their functions, states, and commands in a mutually comprehensible way [10]. SAREF's industry-first design process is notable precisely because it addresses the adoption problem that Shadbolt et al. identified as the Semantic Web's original weak point: an ontology that arises from consensus among the companies that actually have to implement it is more likely to see real deployment than one designed purely for theoretical completeness [3]. SAREF has since been adopted as a foundational reference ontology within the oneM2M IoT standards framework and extended with domain-specific modules for energy, buildings, and other verticals, which is itself evidence that the industry-consensus approach to ontology-building produces artifacts with staying power.

A complementary strand of work has focused specifically on making semantic descriptions lightweight enough for genuinely resource-constrained IoT devices, rather than assuming a device can afford the full SSN/OWL reasoning stack. Bermudez-Edo, Elsaleh, Barnaghi, and Taylor's IoT-Lite ontology explicitly targets this gap: a deliberately minimal semantic model, expressed as a lightweight extension of SSN, that keeps the vocabulary small enough for constrained-device use while still supporting dynamic semantics — the ability to describe a sensor's current state and recent observations rather than only its static capabilities — and the authors report that this reduced footprint materially lowers the

reasoning and storage overhead relative to using the full SSN ontology directly, without sacrificing the ability to query and discover devices semantically [11]. This tension between expressiveness and resource cost recurs throughout the semantic IoT literature and is, in effect, the ontology-engineering analogue of the compression and quantization tradeoffs that the broader embedded-systems literature grapples with for on-device machine learning: richer semantics improve interoperability but cost memory, bandwidth, and processing time that a battery-powered sensor may not have to spare.

Beyond sensing and appliances specifically, the sensor web enablement literature more broadly has worked out what it takes to make sensor networks discoverable and queryable as web resources in the first place. Bröring et al.'s survey of "New Generation Sensor Web Enablement" catalogs the Open Geospatial Consortium's family of standards — Sensor Observation Service, Sensor Planning Service, and related specifications — for publishing, discovering, and tasking sensors over standard web interfaces, and traces how these geospatially oriented standards intersect with, and in places predate, the more recent semantic web ontologies discussed above [12]. Their survey is a useful reminder that semantic interoperability for smart environments did not emerge from a single research thread; it is the confluence of geospatial sensor web standards, general-purpose semantic web ontology engineering, and IoT-specific device vocabularies, each of which developed its own standards bodies and its own vocabulary before beginning to converge.

Taken together, this set of ontologies illustrates a design pattern worth naming explicitly: a smart environment rarely needs one monolithic ontology covering every device class it might ever contain. It needs a small, well-governed core (SOSA's minimal sensing vocabulary, or SAREF's minimal appliance vocabulary) that most devices can afford to implement fully, plus a disciplined extension mechanism for the richer semantics that only some applications need (the fuller SSN model, or SAREF's domain-specific extension modules for energy and buildings). Every ontology discussed in this section reaches that same core-plus-extension structure independently, which is itself a form of convergent evidence that it is the right shape for this problem, rather than an artifact of any one standards body's preferences.

4. Architectures for Semantic Interoperability

Having a shared ontology is necessary but not sufficient for interoperability; something has to actually use that ontology at runtime to annotate incoming data, reason over it, and expose the results to applications, and a substantial body of systems work has focused on exactly this middleware layer. Barnaghi, Wang, Henson, and Taylor's early treatment of "Semantics for the Internet of Things" laid out the architectural case plainly: as IoT deployments scale past a handful of devices, some machine-interpretable representation of what each device measures and how its data relates to other devices' data becomes necessary simply to keep the system navigable, and the paper surveys the early landscape of ontologies, annotation approaches, and reasoning techniques being proposed to meet that need, while being candid that as of the early 2010s most of this work remained at the prototype stage rather than deployed at scale [13].

Making semantic annotation and reasoning fast enough to run against live IoT data streams, rather than only against static datasets, turned out to be its own research problem. Kolozali, Bermudez-Edo, Puschmann, Ganz, and Barnaghi describe a knowledge-based approach to real-time IoT data stream annotation and processing that couples pattern-based feature extraction from raw sensor streams with an ontology-driven annotation pipeline, explicitly designed so that incoming sensor readings can be semantically labeled and made queryable as they arrive rather than only in a batch post-processing step [14]. This real-time framing matters for smart environments specifically: a caregiver monitoring dashboard or a building energy-management system needs semantically annotated data within seconds of an event occurring, not after an overnight batch job, and the annotation pipeline's own processing overhead has to fit inside that latency budget.

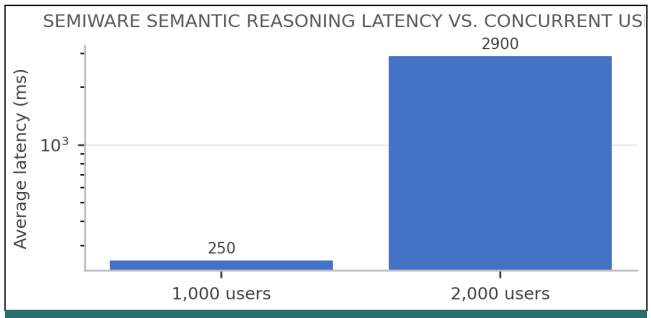


Figure 1: Average semantic reasoning-service query latency under increasing concurrent load, as reported for the SEMIWARE smart-city middleware -- latency stays sub-250ms up to 1,000 concurrent users but rises sharply to roughly 2.9 seconds at 2,000 concurrent users, illustrating the reasoning-layer bottleneck discussed in the Open Challenges section [16].

A second architectural pattern addresses cross-domain reuse: rather than building a bespoke ontology and reasoning pipeline for every new IoT application, can a shared framework of machine-to-machine measurement ontologies and reasoning rules be assembled once and reused across domains? Gyrard, Datta, Bonnet, and Boudaoud's M3 (Machine-to-Machine Measurement) framework takes exactly this approach, combining a cross-domain dataset of ontologies with a rule-based reasoning engine and an accompanying web-based tool for application developers, and the reported evaluation demonstrates the framework's use across genuinely different application domains — the paper's own worked examples span smart home, smart city, and healthcare scenarios — to argue that a shared semantic backbone can meaningfully reduce the per-application engineering effort of building semantic IoT applications from scratch [15]. The M3 framework's central bet is that the hard part of semantic IoT development is not writing OWL syntax per se but knowing which existing ontology, out of the dozens now published, actually fits a given sensor or measurement — a discovery and reuse problem that recurs throughout this literature.

The most recent and most deployment-oriented architecture we reviewed is Goumopoulos and Pliatsios's SEMIWARE, a smart-city middleware that combines a FIWARE-based backbone with modular semantic services — context management, semantic reasoning, and social-relationship management for Social Internet of Things scenarios — built around an OWL2 ontology modeling IoT entities, users, services, and trust-aware social relationships between them [16]. What distinguishes SEMIWARE from most of the earlier architectural proposals reviewed here is that the authors report concrete, load-tested performance figures rather than only a qualitative architecture description: under increasing simulated workload, their semantic annotation service maintained an average processing time of roughly 6.7 milliseconds at 1,000 messages, rising only modestly to about 7.5 milliseconds at 60,000 messages, while their context management service's average response time moved from roughly 25.5 milliseconds at 1,000 requests to roughly 31.7 milliseconds at 60,000 requests [16]. The semantic reasoning service, which is computationally the heaviest of the three, told a less forgiving story: average query latency stayed below roughly 250 milliseconds up to about 1,000 concurrent users, but climbed to roughly 2.9 seconds at 2,000 concurrent users, with the authors reporting a measurable error rate (about 1.8%) once load reached 3,000 concurrent users [16]. Figure 1 visualizes this reasoning-service degradation curve. We read this result not as a weakness specific to SEMIWARE but as an honest and unusually well-quantified illustration of a pattern the rest of the literature mostly asserts qualitatively: semantic annotation and simple context lookups scale gracefully, but OWL-level reasoning over a growing, richly interconnected knowledge graph is the component of a semantic IoT middleware stack most likely to become the bottleneck as a deployment grows from a testbed into a real city-scale system.

5. Ontology Alignment and Cross-Domain Integration

Even a smart environment built entirely out of SSN/SOSA- and SAREF-compliant devices will, in practice, run into a subtler interoperability problem: different application domains, and often different vendors within the same domain, extend these base ontologies in their own ways, define their own domain-specific vocabularies for concepts the base ontologies do not cover, or simply adopt entirely different ontologies for historical reasons. Making two independently developed ontologies interoperate without a full manual rewrite is the problem of ontology matching, and it has developed into a substantial research field of its own, distinct from ontology engineering in the same way that database schema mapping is distinct from database design. Euzenat and Shvaiko's comprehensive treatment of ontology matching lays out the technical landscape — string-similarity and structure-based matching techniques, instance-based matching that compares

how two ontologies' individuals are actually populated, and the alignment representation formats needed to record and reuse a discovered mapping once it has been found — and situates ontology matching as a necessary complement to, not a replacement for, the standardization efforts discussed in the previous section, precisely because standardization can never fully anticipate every domain extension a future application will need [17].

Discovery, not just matching, is itself a nontrivial obstacle: an IoT application developer who wants to reuse an existing ontology first has to find it among the hundreds of published vocabularies covering overlapping domains with inconsistent naming conventions and uneven documentation quality. Vandenbussche, Atemezing, Poveda-Villalón, and Vatant's Linked Open Vocabularies (LOV) project addresses exactly this discovery problem by curating a searchable, quality-checked catalog of published RDF vocabularies and ontologies, along with metadata about how frequently and in what contexts each vocabulary is actually reused across the Linked Data cloud, functioning as a de facto reputation and reuse-tracking system for the ontology ecosystem [18]. For an IoT system designer choosing between, say, three different competing vocabularies for describing energy consumption, a resource like LOV converts what would otherwise be a fairly opaque decision into one informed by observable community adoption, which matters because an ontology's practical interoperability value is almost entirely a function of how many other systems already use it.

At the platform level, the INTER-IoT project's work, described by Ganzha, Paprzycki, Pawłowski, Szmeja, and Wasielewska, tackles semantic interoperability not between individual ontologies but between entire IoT platforms — commercial and open-source IoT middleware stacks that were never designed to interconnect, each with its own internal data model, device registry format, and API conventions [19]. Their approach layers a semantic mediation component on top of existing platform-to-platform integration work, translating between each platform's native data model and a shared semantic representation so that a device registered on one platform can be discovered and controlled from another without either platform's implementation being modified. This platform-level framing is a useful corrective to a tendency in the ontology literature to treat interoperability as purely a device-description problem: in real multi-vendor smart-environment deployments, at least as much friction comes from incompatible platform APIs and data models as from incompatible device ontologies, and a semantic mediation layer has to address both simultaneously to be useful.

Two further threads round out the architectural picture. Guinard, Trifa, and Wilde's resource-oriented architecture for the Web of Things argues for treating every physical device as a first-class, URI-addressable web resource exposing a RESTful interface, which does not by itself solve semantic interoperability but establishes the addressing and access-pattern foundation that semantic annotations are layered onto — a device's semantic description is only useful if the device itself is reachable through a uniform interface in the first place [20]. Serrano et al.'s work defining the service-delivery stack for OpenIoT-VDK makes a related architectural point from the platform-integration side, arguing that interoperability has to be designed as an explicit layered stack — spanning service delivery, data models, and semantic annotation together — rather than bolted on to an existing platform after the fact, and demonstrating the approach on a practical multi-vendor virtualized-device-kit deployment [21]. And Al-Fuqaha, Guizani, Mohammadi, Aledhari, and Ayyash's broad survey of IoT enabling technologies situates semantic-web-based interoperability as one layer within the much larger IoT technology stack — alongside communication protocols, identification schemes, and localization technologies — which is a useful reminder that semantic interoperability, however central to this article's focus, is only ever one piece of what makes a smart environment actually function end to end [22].

6. Open Challenges

Several gaps recur across the literature synthesized here and deserve to be stated plainly rather than smoothed over. First, and most concretely, is the reasoning-scalability problem that SEMIWARE's own load testing makes vivid: semantic reasoning over an OWL2 knowledge graph degrades from sub-250-millisecond latency to multi-second latency somewhere between roughly 1,000 and 2,000 concurrent users in a real deployment, with measurable errors appearing by 3,000 concurrent users [16]. A single smart

home rarely approaches this load, but a smart-city-scale deployment integrating thousands of buildings very plausibly does, and relatively few of the architectures we reviewed report load-tested numbers at all, which makes it hard to know how representative SEMIWARE's curve is of the field generally rather than of one particular implementation choice.

Second, ontology fragmentation remains a genuine, unresolved tension rather than a solved problem papered over by SSN/SOSA and SAREF's existence. Both are widely cited and, in SAREF's case, formally adopted into the oneM2M standard, but the broader landscape LOV catalogs is enormous and still growing, and Euzenat and Shvaiko's ontology-matching techniques exist precisely because standardization on its own has not eliminated the need to reconcile independently developed vocabularies [18,17]. The honest state of the field is that ontology matching is a permanent, ongoing cost of semantic interoperability rather than a one-time engineering task that standardization retires.

Third, the tradeoff between semantic expressiveness and resource cost that motivated IoT-Lite's design is not fully resolved simply by having a lightweight ontology available; a great deal of the semantic-IoT literature still assumes gateway- or cloud-side reasoning rather than genuinely on-device semantic processing, which reintroduces the same "trust the network and the gateway" architecture that motivated federated and edge-native approaches to machine learning in adjacent parts of the IoT literature [11]. How much semantic reasoning can realistically move onto the constrained device itself, rather than being offloaded to more capable infrastructure, remains comparatively under-explored relative to how well the ontologies themselves are specified.

Fourth, evaluation practice across this literature is inconsistent in ways that make cross-paper comparison difficult. Some papers, like Kolozali et al.'s real-time annotation work and Gyrard et al.'s M3 framework, report qualitative architecture descriptions and worked examples without load-tested latency or throughput figures; others, like SEMIWARE, report detailed quantitative results but for a single implementation and workload pattern [14,15,16]. A shared benchmarking methodology for semantic IoT middleware, comparable to the standard benchmarks the broader database and knowledge-graph community has converged on, does not yet appear to exist for this specific application domain, and its absence makes it genuinely hard to say with confidence which architectural choices among those surveyed here generalize best.

Sixth, the interaction between semantic interoperability and security is largely absent from the literature reviewed here despite being obviously relevant to any real smart-environment deployment: a semantic mediation layer that lets one platform discover and control another platform's devices, as INTER-IoT's approach does by design, is also, from a different angle, a mechanism by which access could be granted more broadly than intended if authorization is not modeled with the same rigor as the device semantics themselves [19]. None of the ontologies or middleware architectures surveyed here treat access control as a first-class semantic concern in the way SEMIWARE treats trust-aware social relationships for its specific Social IoT use case, and generalizing that kind of semantically-modeled trust and authorization beyond one application domain remains, to our knowledge, an open problem rather than a solved one [16].

Seventh, and more a matter of adoption than of technology, Shadbolt et al.'s original observation that the Semantic Web's formal machinery outpaced real-world uptake still has real force for IoT specifically [3]. SAREF's industry-consensus development process is, in our reading, the strongest available evidence in this literature that adoption barriers are solvable when standards bodies engage device manufacturers directly rather than designing in isolation, but it remains closer to the exception than the rule across the broader ontology landscape [10].

7. Conclusion

The literature reviewed here supports a fairly specific claim: semantic web technologies — RDF and OWL as a formal substrate, purpose-built domain ontologies like SSN/SOSA and SAREF, lightweight variants like IoT-Lite for constrained devices, and the discovery and matching infrastructure needed to reconcile vocabularies that were never designed to align — together constitute a coherent, increasingly mature architectural pattern for interoperable smart environment integration, not merely a theoretical extension of an older Web Semantics research agenda [4,2]. It inherits the Semantic Web's core interoperability advantage — independently built systems can interpret the same data consistently

because they commit to the same formally specified conceptualization — while the IoT-specific ontology work reviewed here has done the harder job of making that commitment cheap enough for constrained devices and concrete enough for competing manufacturers to actually adopt in a standard like SAREF [10,11].

Making this pattern work at the scale a genuine smart city requires, rather than merely at the scale of a research testbed, depends on treating the open challenges surveyed here as an engineering and governance roadmap rather than a footnote: reasoning-layer scalability under real concurrent load, as SEMIWARE's own numbers make concrete [16]; the permanent, ongoing cost of ontology matching and alignment across a fragmented vocabulary landscape [17,18]; and a benchmarking discipline for semantic IoT middleware that the field has not yet converged on. None of these gaps are reasons to doubt the underlying architectural logic — the consistent convergence of independent standards bodies (W3C, OGC, ETSI, oneM2M) and independent research groups on structurally compatible ontology designs is itself strong evidence that this pattern is sound — but they are the honest state of the evidence, and claims of production-ready, city-scale semantic interoperability should be read against these gaps rather than around them.

8. References

1. Atzori, L., Iera, A., Morabito, G. (2010) The Internet of Things: A Survey. Computer Networks, 54(15): 2787-2805. DOI: 10.1016/j.comnet.2010.05.010

2. Berners-Lee, T., Hendler, J., Lassila, O. (2001) The Semantic Web. Scientific American, 284(5): 34-43. DOI: 10.1038/scientificamerican0501-34

3. Shadbolt, N., Berners-Lee, T., Hall, W. (2006) The Semantic Web Revisited. IEEE Intelligent Systems, 21(3): 96-101. DOI: 10.1109/MIS.2006.62

4. Gruber, T.R. (1993) A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2): 199-220. DOI: 10.1006/knac.1993.1008

5. Studer, R., Benjamins, V.R., Fensel, D. (1998) Knowledge Engineering: Principles and Methods. Data & Knowledge Engineering, 25(1-2): 161-197. DOI: 10.1016/S0169-023X(97)00056-6

6. Bizer, C., Heath, T., Berners-Lee, T. (2009) Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems, 5(3): 1-22. DOI: 10.4018/jswis.2009081901

7. Hogan, A., Blomqvist, E., Cochez, M., D'Amato, C., De Melo, G., Gutierrez, C., Kirrane, S., Labra Gayo, J.E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.-C., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A. (2022) Knowledge Graphs. ACM Computing Surveys, 54(4): 1-37. DOI: 10.1145/3447772

8. Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Taylor, K. (2012) The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Journal of Web Semantics, 17: 25-32. DOI: 10.1016/j.websem.2012.05.003

9. Haller, A., Janowicz, K., Cox, S.J.D., Lefrancois, M., Taylor, K., Le Phuoc, D., Lieberman, J., Garcia-Castro, R., Atkinson, R., Stadler, C. (2018) The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. Semantic Web, 10(1): 9-32. DOI: 10.3233/SW-180320

10. Daniele, L., den Hartog, F., Roes, J. (2015) Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology. Formal Ontologies Meet Industry (FOMI 2015), Lecture Notes in Business Information Processing. Springer, Cham. DOI: 10.1007/978-3-319-21545-7_9

11. Bermudez-Edo, M., Elsaleh, T., Barnaghi, P., Taylor, K. (2017) IoT-Lite: A Lightweight Semantic Model for the Internet of Things and its Use with Dynamic Semantics. Personal and Ubiquitous Computing, 21(3): 475-487. DOI: 10.1007/s00779-017-1010-8

12. Broring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R. (2011) New Generation Sensor Web Enablement. Sensors, 11(3): 2652-2699. DOI: 10.3390/s110302652

13. Barnaghi, P., Wang, W., Henson, C., Taylor, K. (2012) Semantics for the Internet of Things: Early Progress and Back to the Future. International Journal on Semantic Web and Information Systems, 8(1): 1-21. DOI: 10.4018/jswis.2012010101

14. Kolozali, S., Bermudez-Edo, M., Puschmann, D., Ganz, F., Barnaghi, P. (2014) A Knowledge-Based Approach for Real-Time IoT Data Stream Annotation and Processing. Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings), and Green Computing and Communications (GreenCom), and Cyber, Physical and Social Computing (CPSCom): 215-222. DOI: 10.1109/ithings.2014.39

15. Gyrard, A., Datta, S.K., Bonnet, C., Boudaoud, K. (2015) Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation. Proceedings of the 3rd International Conference on Future Internet of Things and Cloud (FiCloud): 9-16. DOI: 10.1109/ficloud.2015.10

16. Goumopoulos, C., Pliatsios, A. (2026) SEMIWARE: A Smart City Middleware Empowering Semantic Interoperability via Social IoT Integration. IoT, 7(3): article 53. DOI: 10.3390/iot7030053

17. Euzenat, J., Shvaiko, P. (2013) Ontology Matching, 2nd edition. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-38721-0

18. Vandenbussche, P.-Y., Atemezing, G.A., Poveda-Villalon, M., Vatant, B. (2016) Linked Open Vocabularies (LOV): A Gateway to Reusable Semantic Vocabularies on the Web. Semantic Web, 8(3): 437-452. DOI: 10.3233/SW-160213

19. Ganzha, M., Paprzycki, M., Pawlowski, W., Szmeja, P., Wasielewska, K. (2017) Towards Semantic Interoperability Between Internet of Things Platforms. In: Internet of Things: Integration, Interconnection, and Interoperability of IoT Systems. Springer, Cham. DOI: 10.1007/978-3-319-61300-0_6

20. Guinard, D., Trifa, V., Wilde, E. (2010) A Resource Oriented Architecture for the Web of Things. Proceedings of the 2010 Internet of Things (IOT) Conference. DOI: 10.1109/IOT.2010.5678452

21. Serrano, M., Nguyen Mau Quoc, H., Le Phuoc, D., Hauswirth, M., Soldatos, J., Kefalakis, N., Jayaraman, P.P., Zaslavsky, A. (2015) Defining the Stack for Service Delivery Models and Interoperability in the Internet of Things: A Practical Case With OpenIoT-VDK. IEEE Journal on Selected Areas in Communications, 33(4): 676-689. DOI: 10.1109/jsac.2015.2393491

22. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M. (2015) Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys & Tutorials, 17(4): 2347-2376. DOI: 10.1109/COMST.2015.2444095

9. Declarations

Funding

This literature synthesis received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability Statement

This article is a literature synthesis and generated no new primary data; all findings discussed are drawn from the cited publications, which readers should consult directly for their own data and code availability statements.

Ethical Approval

Not applicable. This work did not involve human participants, animal subjects, or the collection of primary data, and required no ethical approval.