assistants
Testing per smart home sicure, private e sostenibili

D3.3

Pilot Evaluation

WP3 – Pilot

ASSISTANTS

Assessing Software Security, prIvacy, and SusTAiNability through Testing techniqueS

Due date of deliverable: 28/02/2026 (M27)
Actual submission date: 28/02/2026

Responsible partner: Politecnico di Torino (POLITO)
Editor: Luca Ardito
E-mail address: luca.ardito@polito.it

28/02/2026
Version 1.1

Project funded by the European Union – NextGenerationEU within the PRIN 2022 PNRR programme		
Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium	



The ASSISTANTS project (P2022WEAH7, CUP E53D23016400001) is funded by the European Union – NextGenerationEU under the PRIN 2022 PNRR programme of the Italian Ministry of University and Research (D.D. 1409 of 14/09/2022), Mission 4, Component 2, Investment 1.1.

Authors: Luca Ardito (POLITO)

Approved by: Luca Ardito (POLITO), Principal Investigator

Revision History

Version	Date	Name	Partner	Sections Affected / Comments
1.0	28/02/2026	Luca Ardito	POLITO	All. Final version
1.1	28/02/2026	Luca Ardito	POLITO	All. Revision for publication on the project website

Disclaimer

Funded by the European Union – NextGenerationEU. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Executive Summary

This deliverable reports the experimental evaluation and validation of the ASSISTANTS pilot, carried out in Task T3.3. The pilot, implemented in D3.2 on the Smartotum home-automation system, was evaluated against the requirements and scenarios defined in D3.1, with three goals: verifying that the privacy rules defined by the inhabitants are actually enforced on the devices, verifying that the policy dashboard produces correct policies, and measuring the behaviour of the Privacy Dashboard that drives the whole chain.

The enforcement of privacy rules was tested on a Smartotum camera and on the smart light built with the ASSISTANTS framework. Rules defined through the graphical interface blocked the camera stream and the switching on of the light inside their time window, and both devices behaved normally outside it; the restriction was lifted automatically at the end of the window. Four ways of accessing the camera were verified: from the home network of the Internet provider, where the camera is isolated in a separate network segment; from outside the home, where the stream is reachable only through the Smartotum VPN; from applications running on the gateway; and directly from the local network of the gateway, where the firewall rules set by the privacy manager prevent any access while a rule is active.

The policy dashboard was verified on the generation of XACML policies from natural language and voice, their equivalence with policies written by experts and their robustness to typos, missing words and wrong or ambiguous wording. The Privacy Dashboard was benchmarked in a laboratory deployment connected to a real Smartotum instance: the median API response time is 14 ms; the processing of a consent withdrawal by the dashboard, from the request to the publication of the deny rule to an in-memory DHT proxy, completes in under 1.1 s at the 95th percentile, a figure that excludes the network latency of Smartotum and the application of the rule on the device by the privacy manager; conflict detection takes on average 2.4 ms for 50 rules. All requirements of D3.1 are covered by at least one verification. The successful evaluation completes WP3 and achieves Milestone MS3, *Pilot deployed*.

Table of contents

Executive Summary 3
1 Introduction 5
1.1 Purpose and scope 5
1.2 Relation with other deliverables 5
1.3 Structure of the document 5
2 Experimental Setup 6
2.1 Home-automation environment 6
2.2 Privacy Dashboard environment 6
3 Enforcement of Privacy Rules on the Devices 7
3.1 Test procedure 7
3.2 Smart light 7
3.3 Smart camera 8
3.4 Camera access scenarios 10
4 Verification of the Policy Dashboard 11
4.1 Policies from natural language and voice 11
4.2 Conflict detection 11
4.3 Consent withdrawal 12
5 Operational Benchmarks of the Privacy Dashboard 13
6 Qualitative Feedback from Demonstrations 15
7 Coverage of the Requirements 16
8 Lessons Learned 18
9 Conclusions 19
References 20
Glossary 21

1 Introduction

1.1 Purpose and scope

This deliverable is the output of Task T3.3, *Experimental Evaluation and Validation*, of WP3. It is a report. It describes the experimental setup, the tests performed on the devices of the pilot, the verification of the policy dashboard, the operational measurements of the Privacy Dashboard, the qualitative feedback gathered during demonstrations, the coverage of the requirements and the lessons learned.

1.2 Relation with other deliverables

The evaluation follows the scenarios and measurement points defined in D3.1 and verifies the implementation described in D3.2. Requirement identifiers (FR, NFR, SPR), scenario identifiers (S1–S9) and measurement points (MP-1 to MP-6) refer to D3.1. The Privacy Dashboard is described in D2.3, and the analysis of the Rust software of the pilot with the project tools is reported in D1.3.

1.3 Structure of the document

Section 2 describes the experimental setup. Section 3 reports the tests of the privacy rules on the smart light and the camera, including the four camera access scenarios. Section 4 reports the verification of the policy dashboard. Section 5 reports the operational benchmarks of the Privacy Dashboard and Section 6 the qualitative feedback from demonstrations. Section 7 summarises the coverage of the requirements, Section 8 the lessons learned, and Section 9 concludes the document.

2 Experimental Setup

2.1 Home-automation environment

The device-level tests were performed on a Smartotum installation with its gateway, its local Distributed Hash Table (DHT) and the cloud environment used for remote access through VPN [1]. The ASSISTANTS components described in D3.2 were integrated in this environment:

- the smart light, an ESP32-C3 device running the ASSISTANTS firmware, connected to the Wi-Fi network of the gateway and to its MQTT broker;
- the Smartotum controller and the privacy manager, running as Docker containers on the gateway and started automatically at boot;
- a Smartotum camera, isolated by the gateway in a dedicated network segment.

Privacy rules were defined through the graphical interface and stored in the DHT as `privacy_rule` topics. The integration of the components into the Smartotum ecosystem was straightforward and consistent with the existing architecture [1].

2.2 Privacy Dashboard environment

The Privacy Dashboard was evaluated in a laboratory deployment hosted on a Proxmox virtualisation server running Docker containers orchestrated with docker-compose. The deployment was connected to a real Smartotum instance, so that the consent enforcement pipeline was validated end to end against actual DHT-based communication. The back end is also verified by an automated API test suite (Postman/Newman). The full suite comprises 404 requests and 1,264 assertions; a reduced collection of 161 requests and 317 assertions runs in the continuous integration pipeline, against the complete containerised stack started for each build.

3 Enforcement of Privacy Rules on the Devices

3.1 Test procedure

The components and their interactions were tested with a Smartotum camera and with the smart light [1]. Two privacy rules were defined through the graphical interface:

- the first blocked the command that switches the smart light on in an evening time window, from 21:00 to 22:00, and became automatically invalid from 22:01;
- the second, analogous to the first, blocked the video streaming of the camera from 15:00 to 16:00 and became automatically invalid from 16:01.

The commands were executed first outside the time window, to verify normal operation, and then inside the window. For each test the state of the device topic in the DHT (MP-1) and the state of the Smartotum interface (MP-2) were observed.

3.2 Smart light

Outside the time window the light was switched on and off normally from the interface. Inside the window the light was not switched on. The privacy manager had added the `privacy` and `privacy_until` fields to the topic of the light (Figure 1), and the Smartotum interface detected the condition through the DHT and reported that the actions on the device were temporarily disabled (Figure 2). At the end of the window the normal behaviour was restored without manual intervention.


```
{
  "topic_name": "smart_light_assistants",
  "topic_uuid": "caba8a46-9486-4333-8d80-36cf6ddee387",
  "value": {
    "area_name": "bf673d0c-13ff-4e7c-9169-9a2cf2d28bb4",
    "ip_address": "10.0.1.136",
    "mac_address": "4a:31:b7:3e:46:6c",
    "mode": "manual",
    "privacy": true,
    "privacy_until": "22:00",
    "name": "Luce Intelligente",
    "status": true,
    "temperature": 22
  }
}
```


Figure 1: Topic of the smart light with an active privacy rule (source: [1])

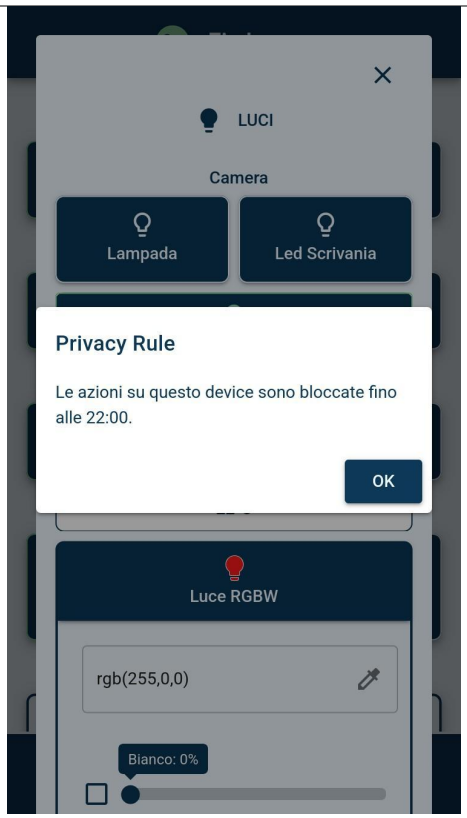


Figure 2: Actions on the smart light blocked by the privacy rule (source: [1])

3.3 Smart camera

Without an active rule the camera showed the stream normally in the Smartotum application (Figure 3). Inside the time window the stream was not started and the application showed an image stating that access was temporarily disabled (Figures 4 and 5). The rule targeting the camera is shown in Figure 6. At the end of the window the function was restored automatically, without changes to the configuration of the device.

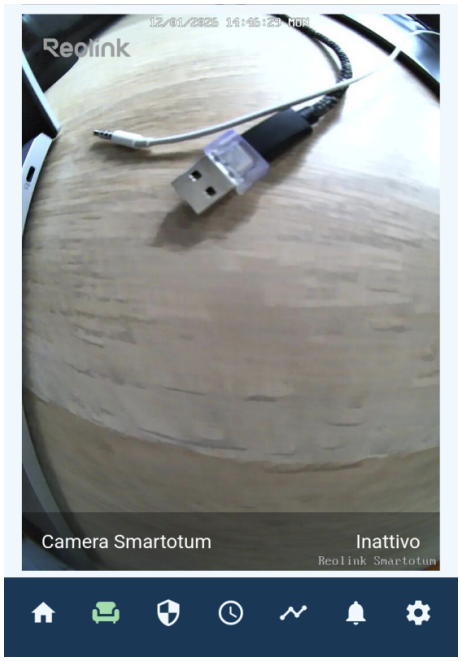


Figure 3: Camera streaming without an active privacy rule (source: [1])

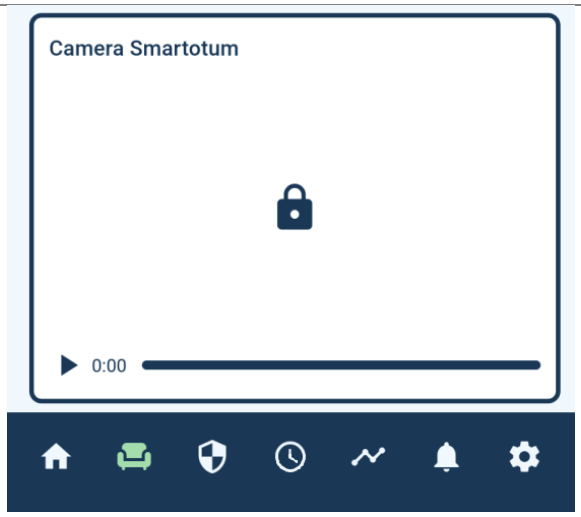


Figure 4: The camera blocked by the privacy rule (source: [1])

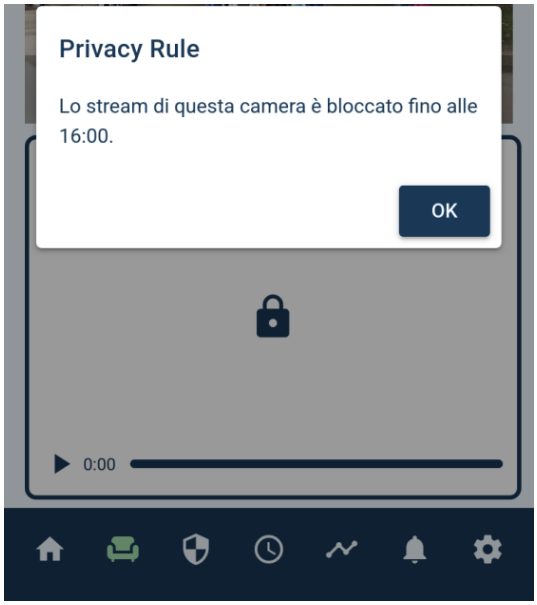


Figure 5: The camera blocked by the privacy rule until the end of the window (source: [1])


```
{
  "topic_name": "privacy_rule",
  "topic_uuid": "a79882ed-5aa0-451b-bd5b-c5147e801d0c",
  "value": {
    "target_topic": "domo_camera",
    "target_uuid": "d6422b30-f800-4be1-8fd4-ccc36a7efcb8",
    "time_start": "15:00",
    "time_end": "16:00",
    "days": [
      "Monday",
      "Wednesday",
      "Friday"
    ],
    "expiration_date": "2026/01/29"
  }
}
```


Figure 6: Privacy rule for the camera in the DHT (source: [1])

3.4 Camera access scenarios

To confirm that the rules hold on every path through which the camera can be reached, four access scenarios were tested [1]. The results are summarised in Table 1.

Table 1: Camera access scenarios and results

Scenario	Access path	Observed behaviour	Result
S1	Local access through the network of the Internet provider	The camera is neither visible nor reachable from devices connected to the home router: it is isolated in a separate VLAN managed by the Smartotum gateway	Passed
S2	Remote access from outside the home	The camera has a local address inside the Smartotum network and is not directly reachable from outside; the stream is accessible only through the Smartotum VPN, so the privacy rules apply to external users too	Passed
S3	Access from applications inside the gateway	Applications running in Docker containers on the gateway respect the privacy rules; the user interface shows the stream as blocked while the rule is active	Passed
S4	Direct access from the local network of the gateway	Devices connected to the local network of the gateway cannot access the camera while a rule is active: the firewall rules set on the gateway by the privacy manager prevent access until the rule ends	Passed

Together with S5 (smart light) and S6 (camera streaming), these scenarios show that a rule defined once by the inhabitant is enforced consistently in the user interface, in the applications of the gateway and at network level.

4 Verification of the Policy Dashboard

4.1 Policies from natural language and voice

The policy dashboard lets inhabitants express a privacy preference in natural language, written or spoken. Language models extract the effect, subject, action, resource, environment and conditions of the preference, and a template builds the corresponding XACML policy [3], which can then be checked and enforced. The approach was explored with several families of models, including Mistral, GPT, Llama and Gemma, and the derivation of usage control policies from natural language was studied in [2].

The verification campaign covered (scenario S7):

- the generation of XACML policies from natural language and from voice;
- the equivalence between the generated policies and the policies written by experts for the same preferences;
- the robustness of the generation to typos;
- the handling of missing words;
- the handling of wrong or ambiguous formulations.

4.2 Conflict detection

Before rules are deployed, the Policy Translation Point checks them for conflicts using Semantic Coloured Petri Nets. Rules are represented as transitions over places describing the state of the device capabilities; simulating the net reveals redundant rules (identical effects that can be merged) and inconsistent rules (opposite effects on the same resource). The Data Controller must resolve inconsistencies before the rules reach the devices. Figure 7 shows the verification report presented in the dashboard for a set of rules on a laboratory light.

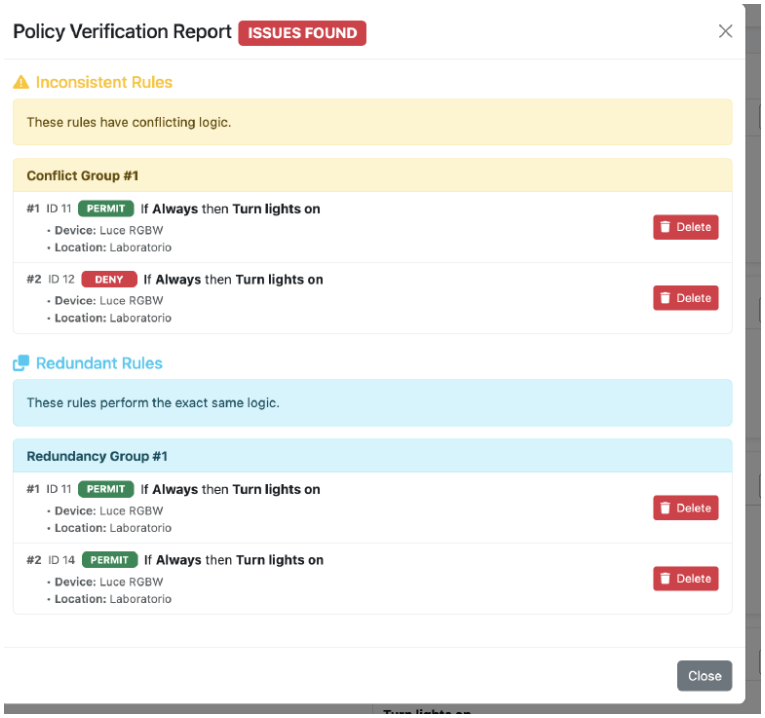


Figure 7: Policy verification report with inconsistent and redundant rules

4.3 Consent withdrawal

The consent withdrawal flow was verified from the Data Subject interface (scenario S8). The inhabitant selects the consents to withdraw for an installed application; when a consent is linked to a home-automation action, the platform creates an automatic deny policy in the selected home and publishes it to the DHT (Figure 8). For the camera manager application, withdrawing the consent to the processing of personal information produces a deny rule for the video recording action.

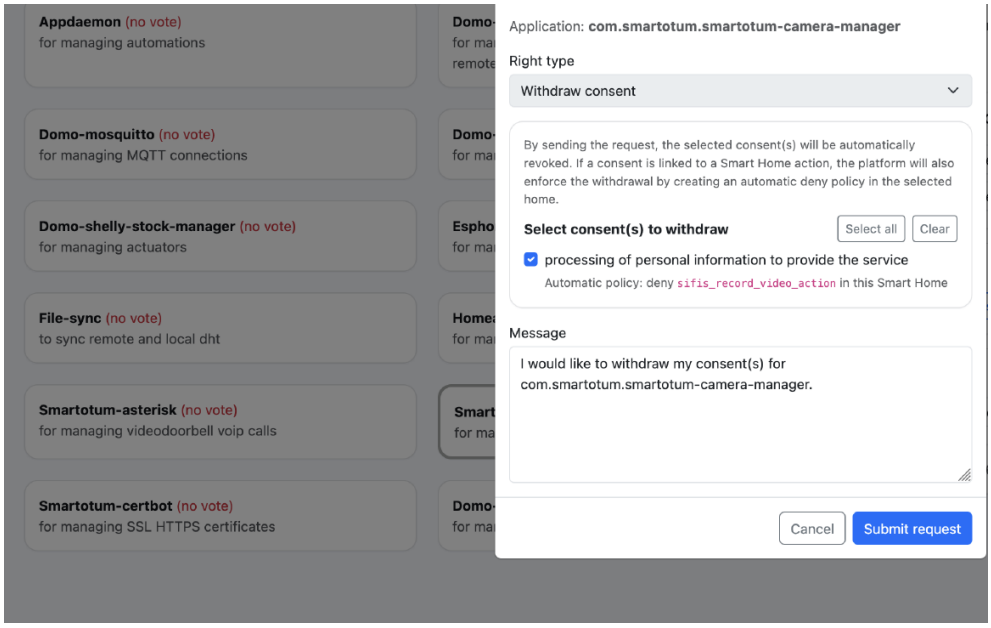


Figure 8: Consent withdrawal with automatic deny policy

5 Operational Benchmarks of the Privacy Dashboard

The five operations most relevant to practical deployment were benchmarked on the laboratory deployment: API response times, consent withdrawal propagation, XACML policy translation, conflict detection with Semantic Coloured Petri Nets and OWL ontology loading (scenario S9, MP-4). The consent withdrawal benchmark used an in-memory DHT proxy, so its latencies measure the processing pipeline of the dashboard without the network delay of Smartotum. The results are reported in Table 2.

Table 2: Operational benchmarks (latencies in milliseconds; n is the number of measured iterations after warm-up)

Operation	n	Median	Mean	p95	Max
API overall (59 endpoint groups)	384	14	177	908	1828
Consent withdrawal (HTTP)	10	867	899	1065	1206
Consent withdrawal (end-to-end)	10	879	908	1072	1212
XACML translation (simple deny)	30	1000	1038	1384	1408
XACML translation (temporal deny)	30	1002	1046	1423	1462
XACML translation (3-subject batch)	30	3040	3188	3558	3609
SCPN analysis (10 rules)	50	0.26	0.28	0.41	0.47
SCPN analysis (25 rules)	50	0.60	0.70	0.91	2.07
SCPN analysis (50 rules)	50	1.58	2.41	2.26	37.06
OWL ontology loading	10	674	669	685	686

The measurements support the following observations:

- **Interactive use (NFR-01).** The median API response time of 14 ms confirms that the large majority of endpoints, such as the operations on consents, privacy notices, impact assessments and questionnaires, respond interactively. The long tail is due to a few heavy endpoints, in particular XACML translation and the consent queries that trigger cascading rule lookups.
- **Reconfiguration (NFR-02).** A consent withdrawal, including rule derivation, policy translation and propagation to the DHT through the proxy, completes in under 1.1 s at the 95th percentile.
- **Policy translation.** Translation time grows linearly with the number of subjects: about 1 s for a single-subject rule and about 3 s for a batch of three subjects, consistent with sequential processing per subject.
- **Conflict detection.** Analysis stays below one millisecond up to 25 rules and below 2.5 ms on average for 50 rules; the isolated maximum at 50 rules is due to garbage-collection pauses. The

practical limit of about 50 rules is therefore a matter of state-space growth for larger sets, not of latency; a typical household with 5–10 devices and 3–4 users is expected to need 15–30 active rules.

- **Ontology loading.** Loading the OWL ontology takes about 670 ms and happens once, when the policy translation component starts.

6 Qualitative Feedback from Demonstrations

The Privacy Dashboard was presented to project stakeholders and test users in informal demonstration sessions. The sessions were not a controlled user study, but they provided useful indications on the privacy opportunities identified in D3.1:

- **Capability-level granularity.** Most participants chose to customise combinations of capabilities rather than accepting the default all-or-nothing configuration, which suggests a demand for fine-grained privacy control.
- **Temporal and contextual policies.** Participants showed considerable interest in time-based and context-based rules, especially for always-on devices such as cameras and speakers.
- **Consent expiry.** Most participants preferred consents with an explicit expiration date and saw periodic re-consent as a natural part of privacy management.
- **Transparent data flows.** The data-flow visualisation of the dashboard improved the understanding of how data are processed, which can support more informed consent decisions.

7 Coverage of the Requirements

Table 3 summarises how each requirement of D3.1 was verified.

Table 3: Verification of the requirements of D3.1

Requirement	Verification	Evidence	Outcome
FR-01, FR-02	S5, S6	Rules defined in the interface and stored as `privacy_rule` topics	Verified
FR-03, FR-04	S5, S6	`privacy` and `privacy_until` fields in the device topics	Verified
FR-05	S5	Light actions disabled in the interface	Verified
FR-06	S3, S6	Stream not started, blocking image shown	Verified
FR-07, FR-08, FR-09	S5, S6	Automatic restoration at the end of the window; rule no longer valid after its end	Verified
FR-10, FR-11	S8	Automatic deny rule published to the DHT; synchronisation with a real Smartotum instance	Verified
FR-12	S7	Generation from natural language and voice, equivalence with expert policies	Verified
FR-13, FR-14, FR-15	S5	Light operated in its modes from the interface; state reflected through MQTT, controller and DHT	Verified
NFR-01	S9	Median API response time 14 ms	Verified
NFR-02, NFR-03	S5, S6, S8	Rules effective inside their window and lifted at its end; consent withdrawal processed by the dashboard pipeline in under 1.1 s (p95, in-memory DHT proxy)	Verified
NFR-04, NFR-05	Setup	Components running locally on the low-power gateway and on the ESP32-C3 device, without cloud services	Verified by design
NFR-06, NFR-08	Setup	Containers started automatically at gateway boot; no change to the Smartotum architecture	Verified
NFR-07	D1.3	Rust firmware and controllers analysed with the project tools	Verified

NFR-09	S9	Full API test suite (404 requests, 1,264 assertions); CI collection (161 requests, 317 assertions) run against the containerised stack; end-to-end validation against a real Smartotum instance	Verified
SPR-01	S1	Camera isolated in a separate VLAN	Verified
SPR-02	S2	Stream reachable only through the Smartotum VPN	Verified
SPR-03	S3	Gateway applications respect the rules	Verified
SPR-04	S4	Firewall rules on the gateway block direct access	Verified
SPR-05	S8	Deny rule for the action linked to the withdrawn consent	Verified
SPR-06, SPR-07, SPR-08	S8, S9	Smartotum token verification, role-based access, audit of consent and rule changes	Verified

8 Lessons Learned

The pilot and the development of the Privacy Dashboard produced lessons that are useful beyond the project.

Integration across phases and projects. Developing the dashboard in independent phases produced well-modularised subsystems: the Policy Translation Point, for instance, is a self-contained, reusable component. The price is the integration effort when the phases meet a concrete platform. The practices that proved effective are: specifying API contracts and data models early and validating them against the real system; planning integration as an explicit activity; keeping end-to-end tests across all components in continuous integration; and versioning interfaces and data contracts. On the Smartotum side, using the DHT as the only integration point allowed the new components to be added without changing the platform.

A minimal consent-enforcement pipeline. The pilot validated a pattern that can be reused on any home-automation platform exposing device-level control APIs: a high-level rule language (who, what, on which resource, allow or deny, under which conditions); a translation layer towards the policy format of the enforcement system; a connection to the device platform; and conflict detection before the policies reach the devices.

Consent enforcement goes beyond the user interface. Withdrawing a consent with one click satisfies the GDPR requirement [4] that withdrawal be as easy as granting consent, but enforcing it on the devices requires state management, retries when devices are offline and coordination with the platform. Data already recorded on a device or sent to third-party services requires separate deletion mechanisms.

Privacy documentation has a cost. Even with a WYSIWYG editor and structured templates, producing a legally sound privacy notice took a data controller without legal training about 4 to 8 hours per application.

Policy languages and formal checks. XACML provided a standard, expressive policy language with open-source tools, at the cost of verbosity: the translation layer amounts to about 2,000 lines of Python. Semantic Coloured Petri Nets provided rigorous conflict detection that is fast for residential rule sets.

Scope of the evaluation. The evaluation was carried out on a Smartotum installation and in a laboratory deployment connected to a real Smartotum instance, with informal demonstrations rather than a controlled user study. Smartotum plans to integrate the Privacy Dashboard into its production platform, which will make a study with real households, data controllers and data protection officers possible.

9 Conclusions

The evaluation showed that the ASSISTANTS components work together in a real home-automation system. Privacy rules defined by the inhabitants were enforced on a camera and on a smart light inside their time windows and lifted automatically at their end; the camera was protected on all four access paths, from the isolation in a dedicated network segment to the firewall rules set by the privacy manager on the gateway. The policy dashboard was verified on the generation of XACML policies from natural language and voice, and its conflict detection identified inconsistent and redundant rules before deployment. The Privacy Dashboard responded interactively, and its pipeline processed a consent withdrawal up to the publication of the deny rule in about one second; this measure, taken with an in-memory DHT proxy, does not include the network latency of Smartotum or the application of the rule on the device, which the privacy manager performs at its periodic check.

All the requirements of D3.1 were verified. The pilot demonstrates the complete chain that the project set out to build: from the user, through the interface and the policy, to the decision, the enforcement on the device and its verification. With this deliverable WP3 is completed and Milestone MS3, *Pilot deployed*, is achieved.

References

[1] Smartotum, *Specialist consultancy service in home automation for the implementation of security policy enforcement mechanisms in smart home systems*, final report for the ASSISTANTS project, signed by D. De Guglielmo.

[2] Alajramy, Simoni, Rasori, Saracino, Mori, "On-device derivation of IoT usage control policies", *Future Generation Computer Systems*, 2026.

[3] OASIS, *eXtensible Access Control Markup Language (XACML) Version 3.0*, OASIS Standard.

[4] European Parliament and Council, Regulation (EU) 2016/679 (General Data Protection Regulation), 2016.

Glossary

Acronym	Meaning
API	Application Programming Interface
DHT	Distributed Hash Table
GDPR	General Data Protection Regulation
MP	Measurement Point
OWL	Web Ontology Language
SCPN	Semantic Coloured Petri Net
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
XACML	eXtensible Access Control Markup Language