Fursight: A Fully Integrated Solution for Pet Monitoring via CPS and IoT

António Santos[†, 1], João Araújo[†, 1], Jorge Sousa[†, 1], José Castro[†, 1], Guilherme Sequeira[†, 1], Miguel Montes[†, 1], Pedro Ramalho[†, 1], Ricardo Inácio[†, 1], Vítor Cavaleiro[†, 1], Rui Pinto[1,2], Gil Gonçalves[1,2]

Dept. of Informatics Engineering, Faculty of Engineering, University of Porto, Porto, Portugal[1]
SYSTEC, ARISE, Faculty of Engineering, University of Porto, Porto, Portugal[2]
Email: {up202008004, up202004293, up202006140, up202006963, up202004648, up202007516, up202004715, up202302742, up202004724}@up.pt
{rpinto, gil}@fe.up.pt
[†]These authors contributed equally to this work.

***Abstract*—The paper proposes a novel, fully integrated intelligent system that incorporates the Internet of Things and Cyber-Physical Systems technologies to streamline the routine of monitoring household pets' day-to-day life and well-being. By leveraging this system, users become keenly aware of the overall disposition of their pets, allowing for a multimodal data-driven approach to responsible care. Several data collection devices are leveraged to collect information about the current circumstances of a pet, resulting in derivable, up-to-date insights available to the user on the web. Several use cases were used to validate this approach, exemplifying the usefulness of a modular pet-care system that is customisable to varying needs and can be seamlessly deployed locally or to the cloud, ensuring well-being at all times. By leveraging current animal monitoring efforts with new-age data collection technologies, Fursight represents a robust solution to unobtrusive pet care, combining the Internet of Things and human-centred design, to reach optimal care even in the prevailing fast-paced mode of life.**

Keywords-IoT; CPS; pet monitoring; self-deployed applications.

I. Introduction

Taking proper care of a pet is a great responsibility. Several factors need to be accounted for at all times regarding the health and well-being of the animal, which could be divided across diverse contexts that must be harmoniously reconciled. The evolving dynamics of human-companion animal relationships are influenced by modern societal trends, including increased emotional attachment, changing family structures, and external factors such as remote work adoption and the COVID-19 pandemic. Research suggests that while some pet owners experience time constraints due to demanding work schedules, others have adjusted their lifestyles to accommodate more interaction with their pets [1].

Wearable technology has emerged as a key enabler in pet monitoring solutions, providing insights into location tracking, health metrics, and behavioural patterns. These devices, such as GPS collars, heart rate monitors, and activity trackers, are widely used for monitoring pets' well-being and have been increasingly integrated with IoT and smart home systems [2]. However, as demonstrated with *Elizabethan collars* [3], artefacts placed around the throat contribute to distress and discomfort. Adversely, the ever-growing array of options for circumstance monitoring in data collection devices commonly used in Internet of Things (IoT) solutions (e.g., smart room sensors) lay the groundwork for a non-invasive and easily implementable option towards animal companion care. With all of this, an application that takes advantage of the availability of connected systems would be extremely valuable, allowing for continuous and accessible monitoring even when pets need to be left unattended.

Recent advancements in Cyber-Physical Systems (CPS) have facilitated their integration into smart home environments, enhancing automation, security, and remote monitoring capabilities. These developments have led to the widespread adoption of intelligent systems for pet care, enabling real-time monitoring, interactive communication, and adaptive behavioural analysis [4].

In this context, the authors propose a solution, designated Fursight [5]. This solution conducts all its sensing at the Edge, leaving analysis and processing to the central hub, leveraging wireless connectivity to transmit the collected data. By aggregating environment data, valuable insights can be provided to further enhance several care processes. In this sense, different *Arduino Nano* flavours, linked to diverse sensors and actuators, connect to a *Raspberry Pi Model 4* [6] to build a cohesive system which effectively exchanges data across all components. Through `Node-RED` [7], workflows for automatic data processing can be defined, allowing for seamless deployment of different settings by simply swapping configuration files. Hence, such a system can be deployed in any home division where pets may be settled, either locally or online, to streamline the care processes based on insights derived from data. Information is then stored securely, and made available through a web interface that displays to the user visualisations of the sensed environment. Through the installation of an integrated container [8], all the necessary resources are automatically loaded, easily enabling full functionality.

This paper is organized as follows: this Introduction Section I, which outlines the challenges, motivations, and objectives of this work, is followed by the Related Work Section II, which highlights fundamental in-domain research, including a critical analysis of existing solutions to underscore the necessity of this approach. The Proposed Solution Section III elaborates on the proposition, detailing the solution architecture and implementation process, along with the technologies employed. In the Evaluation Section IV, the results of the carried-out tests are presented, incorporating assessment methodologies, and a thorough analysis of the outcomes. The Discussion Section V provides a breakdown of the findings, emphasizing the benefits of the solution while addressing its limitations and identifying areas for potential enhancement. The paper concludes with a Conclusion and Future Work Section VI, which highlights the key contributions and achievements of the study and outlines directions for prospective endeavours.

II. Related Work

In this section, the background that serves as the foundation of the proposed solution is presented. Firstly, related technologies and similar solutions in the current literature are analysed, subsequently showcasing the current gaps addressed in the proposed solution. Presently, most solutions are based on the same set of core technologies, as they are long-standing standards in the field of home automation and connectivity.

The advancements in communication technologies, alongside the decrease in complexity of home automation and monitoring systems, allow for specific solutions that exploit the continuous availability these techniques provide. Thus, several works have been developed recently, to harness these advantages in the context of pet monitoring and care. These are divided across several modalities, such as: wearables that continuously monitor vitals and geolocation; cameras that constantly provide image or video feeds; automatic feeders and water dispensers based on actuators; software apps and platforms, that provide relevant information and aggregate data to deliver insights. For a more holistic assessment of such approaches, refer to the survey about pet care technology [9].

A. Wearables for Continuous Monitoring

Most health and vitals monitoring solutions for pets rely on wearable devices, typically attached via a collar or specialized accessories. For example, Reyu and Princess [10] propose a wearable system that tracks heart rate, temperature, activity levels, and location. However, research suggests that such devices may cause discomfort if they are too obtrusive [3].

In addition to the wearable, their system includes a monitoring application designed to aggregate and display collected data. While no specific mobile architecture is specified, they plan to provide a web-based platform to ensure broader compatibility across devices. Their approach also incorporates machine learning techniques to analyze trends and predict potential health issues, enhancing proactive pet care.

B. Multimedia Capturing Technologies

In the vision modality, the use of image and video capturing technology provides owners with an opportunity to continuously supervise their pets, from any place from which connectivity is available. However, a continuous acquisition of such data results in vast amounts of storage space occupied. Furthermore, only a portion of the recorded events are relevant to the owners, as the majority will consist of mundane activities. This issue is further amplified by the limited resources commonly inherent to IoT devices, which also indirectly pose security risks, as secure and complex encryption may not be feasible, relying on simpler methods instead [11]. Thus, given the nature of the service, if the home-based video or image data must be sent to the internet, a secure connection must be ensured.

C. Automatic Feeders and Water Dispensers

The task of automating the feeding of pets, ensuring they are well-nourished while taking into consideration their specific needs, has been a long-standing effort. It allows owners to ensure the welfare of their pets, even under time or mobility constraints. Advancements in communication technologies, even enable an intelligent feeding process, where food ration sizes are tailored to the pet, based on physical factors and energy requirements [12]. This approach employs a mobile application, paired with a social media account, to remotely control the feeder, schedule feedings, or review feeding history. The benefits can also stem from the possibility of proactive nourishment, given environmental data. For example, if the temperature increases, the water delivery rate could increase as well [13]. However, this association with mobile applications, inadvertently reduces the scope of the install base of the service, as a forced addition of social features may lead to technological anxiety [14], showcasing the need for advanced and digitally integrated, but local, private solutions.

D. Software-based Solutions

Lastly, software-based apps and platforms, developed to either complement the previously mentioned solutions or as standalone services, to aid in pet monitoring and care, are wide-ranging. Some examples that follow the previous procedures are companion apps for live feed cameras and automatic feeders. Some integrate popular AI assistant technology, such as *Google Assistant* [15], to streamline some tasks. Standalone apps that do not require external tools can be valuable, as they reap the benefits brought by mobile computing. They allow users to log veterinary appointments, medications, and vaccinations, or to set reminders for boosters. Other apps allow activity and behaviour tracking, to enhance the level of detail and quality of the care provided. Social features allow for exchanging relevant knowledge, such as training and grooming tips, promoting pet well-being [16], although such focus on applications can lead to negative perception [14], as already mentioned.

III. Proposed Solution

In this section, the methodology behind the proposed approach is presented, alongside the development and implementation details undertaken to reach the final solution. It tries to solve several problems using a package that encompasses features related to the modalities presented in the Related Work Section II. Figure 1 details the overarching design and architecture of the implemented solution, where two Arduino

 ISBN: 978-1-68558-236-4

(X and Y), connected to two submodels each (sensor and actuator), send gathered raw data to the local central hub via the Internet protocol suite (TCP/IP) [17]. An advanced submodule, specifically a serial camera, can be connected to the hub. Using `RabbitMQ` [18], this hub can either send the processed, gathered data to a local, self-hosted server, or to the cloud, to as many servers as it needs, given specific requirements. The engineered architecture was tailored to provide every functionality coherently while also allowing for a distributed approach toward reliance and scalability. It is arranged across three key domains: Physical, Middleware and Application.

A. Physical Layer

The Physical layer is present at a local level in each client setup. It contains several submodules serving different monitoring and actuating purposes, connected to a central hub for processing and communication. While the central hub works as a hotspot using the Dynamic Host Configuration Protocol (DHCP) in the local network, it is also able to connect securely to servers via TCP/IP [17]. The submodules are comprised of sensors and actuators connected to an *Arduino*, using serial protocols, such as the Inter-Integrated Circuit (I2C) [19], SenseWire (I3C) [20], Serial Peripheral Interface (SPI) [21], and Universal Asynchronous Receiver-Transmitter (UART) [22] protocols for linkage. These, in turn, act as the physical hub for the setup placed on each desired division. Multiple of these, spanning different modalities, can be connected to the central hub to provide a more sophisticated monitoring scheme, enabling multiplicity, as each physical device can be easily tracked and referred to using the identification system. The data collected by the sensors is processed in the *Arduino Nano* modules, which are then sent, via TCP/IP [17] over Wi-Fi, to the local *Raspberry Pi 4 Model B* [6] central hub, using the Message Queue Telemetry Transport (MQTT) [23] broker, implemented via `RabbitMQ` [18]. Additionally, more advanced sensors, such as serial cameras, can be installed directly into the main hub, also interfaced via SPI [21], and streaming in real-time via the User Datagram Protocol (UDP) [24]. If this data must be sent online, a secure connection must be ensured, given the sensitivity of the matter (such as household data). Some practical examples consist of movement and proximity detectors, litter box usage tracking, smart feeding, and audiovisual active monitoring. With this modular setup, the flexibility of the solution allows it to be implemented in almost every possible home environment.

B. Middleware Layer

The Middleware servers as the centralized processing unit of all the sensed raw local environment data. Thus, it serves as the central hub of the services, which aggregates all the employed physical hubs (i.e., separate physical hubs each in a different room of the house). The scripting capabilities of the `Python` language are employed, given its relevance for data science, and strong support of related libraries [25]. To exchange information with all the locally present submodules, `RabbitMQ` was chosen for the message broker [18]. This technology provides a robust topic-based messaging system, originally implementing the Advanced Message Queuing Protocol (AMQP) [26], later supporting more standard protocols such as MQTT [23]. Each active sensor is assigned an identifier, which then is used as part of the message topic sent through the network. By using message topics with `RabbitMQ` [18], it is ensured that data from specific sources reaches only desired ends while also identifying the modules it went through. Each *Arduino* also has an identifier assigned, which it adds to the message topic, that already contains the identifiers from the sensors. Here, the central hub acts as the core for data processing, so that it can be stored, to then be analysed and converted into valuable knowledge regarding the pet.

For a more private and local approach, the user is able to use their machine to act as a self-hosted server. In this case, the communication is still made through an MQTT [23] broker, ensuring a fast and lightweight transmission, although not very secure, which is not problematic given the isolated facet of the exchange. Before storage, a local implementation of this layer would have a central hub (*Raspberry Pi Model 4* [6]), adding its identifier to the message topic, already containing the ones from each physical hub module the message passed through. However, if the user prefers an always-available solution, then the server can be deployed online with ease, ensuring secure communication via transport protocols like Secure Sockets Layer (SSL) and Transport Layer Security (TCL). This way, it is possible to distribute the load through several servers, or even to keep replicas of important data. In this case, another identifier is added to the complete chain of identifiers, specifying which local client the message is coming from. This process is shown, in a simplified manner, in Figure 1. If this second route is taken, then contents sent to the cloud are pipelined through `Node-RED` [7] in order to streamline data handling. This tool abstracts this process to a higher level, making it useful in systems where inputs may come from diverse sources. It not only allows data to be accessed in real-time but also easy implementation of custom logic for transforming raw sensor data into structured outputs. It is instantiated in the central hub, and not only receives input through MQTT [23] from sensors and servers but also performs specified transformations that support future analyses, such as moving averages for time series. Furthermore, different pipelines can be employed by simply switching the deployed to a JavaScript Object Notation (JSON) configuration.

Ultimately, this layer becomes optional for small-scale, low-cost processing contexts in which there is, for example, only a single-tracked environment. In this case, data can be sent directly to the application layer from the physical hub.

C. Application Layer

Finally, the Application layer comprises both the backend and frontend of the proposed approach, providing the logic and presentation using the processed data. In the backend, *Supabase* [27] was employed in a Backend as a Service (BaaS)

 ISBN: 978-1-68558-236-4

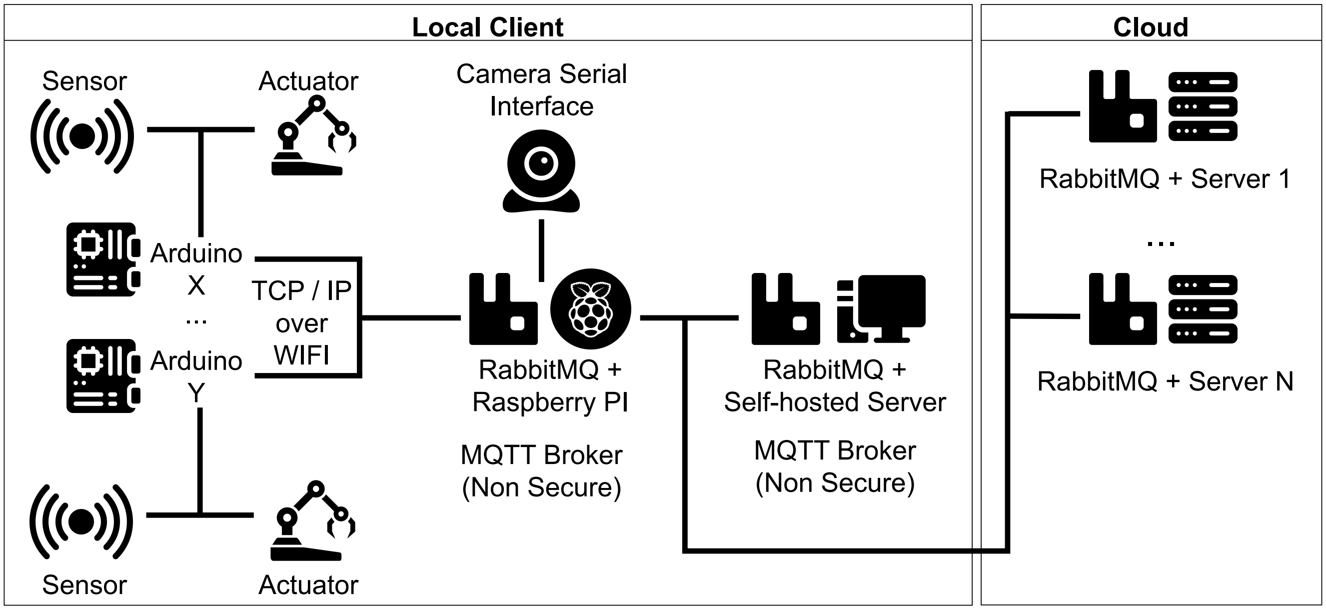

Figure 1. Architecture of the solution.

paradigm, as an authentication-enabled database, to store user-specific location and specification data. Communication at this layer is still made using the previously established `RabbitMQ` settings. Through *Supabase* [27], information related to each user, about the location of service deployment, hardware specifications, and sensor configuration, is saved on different relational tables. To make this data seamlessly accessible, a RESTful API [28] was developed, allowing for CRUD operations on said data (create, read, update, delete). It is then used both on internal Python scripts and via the JavaScript *fetch* API to display extracted knowledge and insights to the end user. Regarding authentication, several options are available so that end users may choose their preferred method. If it is the first time the user uses the system, a *sign-up* portal is available; in case the user already has an account, then it may use their email, phone number, or unique-use code to *sign in*. Also, an OAuth-based alternative [29] is available.

Finally, the User Interface (UI) must be intuitive to the broadest demographic possible, given that the pet-owner profile differs significantly by both ethnicity [30] and age [31]. To that end, the `shadcdn` collection was employed to achieve a simple and consistent presentation. It allows for a quick design process, by simply selecting which UI elements to display. Furthermore, `TailwindCSS` was used to define the remaining styles, which easily integrates the previously mentioned components. For the programming language and framework, `TypeScript` and `Next.js` were chosen, respectively. `TypeScript` enhances the overall experience by enabling a static typing system, easing the integration process of the different elements in the system, and a streamlined refactoring routine. The `Node.js` framework, being `React`-based, provides Server-Side Rendering (SSR), ensuring the application is fast and adheres to the best Search engine optimization (SEO) practices [32], [33]. It also provides automatic code splitting, executing only the needed JavaScript for each page. This advocates a coding practice in which each page becomes a self-contained package, encapsulating its logic and state, and fostering scalability.

The connection between the backend and the frontend components of the Application layer is done through the `fetch` API to execute the requested methods and later display the retrieved responses as diverse visualizations (e.g., line graphs) implemented via the selected and imported components. This provides an interface that is tailored to the modules and submodules each user manages in its home environment, cohesively. It is also through this API that users interact securely with the authentication system.

IV. Evaluation & Results

After designing and developing the complete solution, its applicability was tested through several use cases to assess how it behaves given different real-world conditions. It is of utmost importance that the solution is robust to the variability of the environment it may eventually be installed into, and that it can handle a vast array of different scenarios, proving the universal adaptability and appeal of the approach. However, given the experimental nature of this work, different testing scenarios were implemented using prototypes and scale models as a way to simulate likely circumstances where this solution may be inserted into, proving its practicability more abstractly. These test scenarios were based on a cat monitoring situation, thus comprising different cat care aspects. It should be noted that, based on the versatility of the approach, if the apparatus is applicable in a scale model of a given setting, then it should be, theoretically, also applicable in the circumstances it hopes to simulate.

As previously stated, end users can simply follow a short list of steps to get the solution running locally, by cloning the repository, and executing each component in a containerised manner, automatically. Given that the physical components are correctly connected (as we are leveraging prototype-oriented hardware), in theory, the solution should be able to correctly capture relevant environment data about activities carried out by the pet, or to the pet itself. In this sense, Figure 2 contains an example of a graph visualization accessible through the web application.

A. Pet Monitoring

Room monitoring is an indirect but essential tracking method for pet care applications. To measure the temperature and humidity of the room, a tracking submodule was implemented, powered by a Temperature & Humidity Sensor. Furthermore, environmental healthiness is also measured through a Carbon Monoxide Sensor [34]. In addition to this, a live sound feed of the environment was achieved through a Sound Sensor [35], allowing the solution to register peaks of sound activity that could indicate distress calls. The application was able to display both environmental and sound activity through time-series graphs.

B. Litter Box Activity

To test the applicability of a litter box activity tracking feature, a Soil Moisture Sensor [36] was employed. This sensor was placed inside a small box containing a common household brand of cat litter, which was watered periodically. The data throughput allowed the solution to create a visual representation of how moist the soil was at a given time, through a time-series graph, and effectively highlight litter box activity even through repeated usages in short periods. Moreover, a more elaborate testing procedure regarding this specific behaviour was set forth, in which the footage captured from a serial camera would be used in tandem with the previously mentioned sensor outputs as a way to verify litter occurrences in a multimodal manner. If the live feed displayed significant differences comparatively to a record of an empty litter box, this data could be cross-referenced with moisture sensor data to ensure a verifiable reading of activity.

C. Automatic Pet Feeding

To simulate an automatic pet feeding feature, a Pressure Sensor [37] was utilized and tested against different levels of touch pressure to ascertain if the presence of the pet could be used as an indicator that food should be served. This mechanism could be subsequently implemented in tandem with a feeding actuator (e.g., a small motor connected to a food box door). The application was able to register touch activity and represent it in a time-series graph.

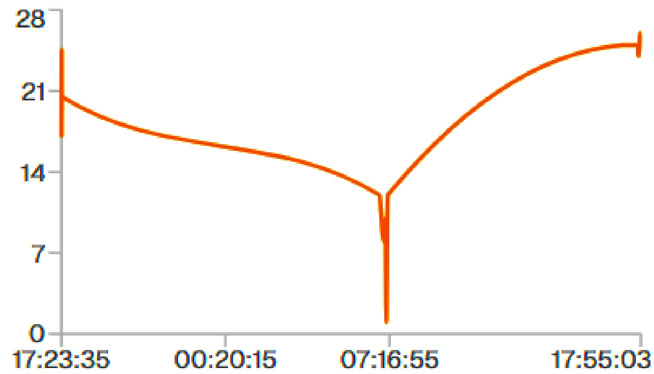

Figure 2. Simple graph visualization of the temperature in Celsius by time of day, present in the web application. Displays a sharp decline during nighttime.

V. Discussion

Several potential benefits surfaced when implementing use cases based on the outlined proposal. As evidenced by the conclusions drawn from the assembled scenarios, Fursight is a rather adaptable system given its modular nature, which also contributes to its scalability factor, as shown by the successful construction of scale models, suggesting applicability in full-scale settings. Furthermore, this ease in implementing different arrangements showcased how streamlined the deployment process is even for end users, showcasing its possible adaptability in other contexts, such as pet hotels or veterinary clinics. In these cases, even against higher setup complexity, Fursight allows each client to manage their system comprehensively through the web application. The developed solution not only allows for multi-faceted and comprehensive monitoring of pet well-being but also permits doing it in real-time via easily interpretable visualizations. Additionally, the non-invasive nature of the solution was upheld throughout development, given that no appliance needed to be set directly on the pet.

Nonetheless, some shortcomings also came to light. Given the experimental nature of the solution, strict bounds were placed on the planned implementations, restraining their faithfulness to their real-world counterparts. For the same reason, a rigid emphasis is put on a correct physical setup. Moreover, since the devices used for the physical components stem from a prototypal nature, the showcased sensibility was sometimes inappropriate, as seen in the pressure sensor, which ought to be triggered by non-pet activities. Overall, although the practical value of the solution was showcased, it would not be a reasonable commercial solution in its current state.

VI. Conclusion & Future Work

This project demonstrated a novel integrated solution for pet monitoring using IoT and CPS, called Fursight. This modular approach to pet care seamlessly integrates several sensors (and actuators) to monitor pets, alongside their involving environment, without the need to hook attachments to their bodies. The modular architecture, comprising physical, middleware, and application layers, allows for adaptive deployment in various home configurations. While promising results were produced during simulated experiments, several limitations surfaced, mainly due to the experimental nature of the setup.

For future work, given the modularity aspect of the solution, it would be beneficial to further integrate more complex submodules, such as making the live feed constantly and irrevocably available to the end user, while also enhancing accuracy in activity detection. In a commercial context, it would be best to provide a more streamlined setup process by devising a simpler integrated standalone service, combined with a more robust, sellable physical package, which could comprise diverse key components required for the full operational capability of the solution, as well as modular expansions for extra functionality. Ultimately, Fursight represents a robust solution to unobtrusive pet care, combining IoT and human-centred design, to reach optimal care even in the prevailing fast-paced mode of life.

Acknowledgment

This work was financially supported by project openZDM – "Open Platform for Realizing Zero Defects in Cyber Physical Manufacturing" - GA nº. 101058673, funded by the European Commission under the Horizon Europe program. The authors gratefully acknowledge the support provided by the Foundation for Science and Technology (FCT/MCTES) within the scope of the Associated Laboratory ARISE (LA/P/0112/2020), and the R&D Unit SYSTEC through Base (UIDB/00147/2020) and Programmatic (UIDP/00147/2020) funds.

References

[1] K. J. Scoresby *et al.*, "Pet Ownership and Quality of Life: A Systematic Review of the Literature," *Veterinary Sciences*, vol. 8, no. 12, 2021, ISSN: 2306-7381. DOI: 10.3390/vetsci8120332.

[2] A. Mitek, D. Jones, A. Newell, and S. Vitale, "Wearable devices in veterinary health care," *Veterinary Clinics: Small Animal Practice*, vol. 52, no. 5, pp. 1087–1098, 2022.

[3] Y. Shenoda, M. P. Ward, D. McKeegan, and A. Fawcett, "The cone of shame: welfare implications of Elizabethan collar use on dogs and cats as reported by their owners," *Animals*, vol. 10, no. 2, p. 333, 2020.

[4] Y. Chen and M. Elshakankiri, "Implementation of an iot based pet care system," in *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2020, pp. 256–262. DOI: 10.1109/FMEC49853.2020.9144910.

[5] A. Santos *et al.*, "Fursight's Github Repository," [Online]. Available: https://github.com/FurSAIght/FurSight (visited on 01/14/2025).

[6] P. D. P. Adi *et al.*, "Optimization and Development of Raspberry Pi 4 Model B for the Internet of Things," in *2023 IEEE 9th Information Technology International Seminar (ITIS)*, IEEE, 2023, pp. 1–6.

[7] R. K. Kodali and A. Anjum, "IoT based home automation using node-red," in *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, IEEE, 2018, pp. 386–390.

[8] J. Nickoloff and S. Kuenzli, *Docker in action*. Simon and Schuster, 2019.

[9] M. Tauseef, E. Rathod, S. M. Nandish, and M. G. Kushal, "Advancements in Pet Care Technology: A Comprehensive Survey," in *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)*, Mar. 2024, pp. 1–6. DOI: 10.1109/ICDECS59733.2023.10503555.

[10] S. Nithin Wilfred Reyu and J. Pearly Princess, "IoT based Pet Health Monitoring System for Sick Pets," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Jun. 2024, pp. 1648–1652. DOI: 10.1109/ICAAIC60222.2024.10574965.

[11] I. Rozlomii, A. Yarmilko, and S. Naumenko, "Data security of IoT devices with limited resources: challenges and potential solutions.," in *doors*, 2024, pp. 85–96.

[12] Y. Quiñonez, C. Lizarraga, R. Aguayo, and D. Arredondo, "Communication architecture based on IoT technology to control and monitor pets feeding," en, *JUCS - Journal of Universal Computer Science*, vol. 27, no. 2, pp. 190–207, Feb. 2021, ISSN: 0948-6968, 0948-695X. DOI: 10.3897/jucs.65094.

[13] W. P. Rey and K. W. J. D. Rey, "AZKALS: Revolutionizing pet care with an IoT-powered smart dog cage management system," in *2024 3rd International Conference on Computer Technologies (ICCTech)*, IEEE, 2024, pp. 48–54.

[14] J.-W. Kang and Y. Namkung, "The role of personalization on continuance intention in food service mobile apps: A privacy calculus perspective," *International Journal of Contemporary Hospitality Management*, vol. 31, no. 2, pp. 734–752, 2019.

[15] V. Prithviraj and K. Sriharipriya, "Smart Pet Feeder System Based on Google Assistant," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, IEEE, 2022, pp. 1–6.

[16] M. Sharma and R. Gautam, "Evolving Pet Care Landscape: Exploring Digital Marketing, Educational Tech, and User Experience Trends," *Advances in Urban Resilience and Sustainable City Design*, vol. 14, no. 12, pp. 13–34, 2022.

[17] V. Cerf and R. Kahn, "A protocol for packet network intercommunication," *IEEE Transactions on communications*, vol. 22, no. 5, pp. 637–648, 1974.

[18] G. M. Roy, *RabbitMQ in depth*. Simon and Schuster, 2017.

[19] J. Mankar, C. Darode, K. Trivedi, M. Kanoje, and P. Shahare, "Review of I2C protocol," *International Journal of Research in Advent Technology*, vol. 2, no. 1, pp. 474–479, 2014.

[20] M. Alliance, "I3C," 2025, [Online]. Available: https://www.mipi.org/specifications/i3c-sensor-specification (visited on 01/15/2025).

[21] A. Meroth and P. Sora, "Serial Peripheral Interface (SPI)," in *Sensor networks in theory and practice: Successfully realize embedded systems projects*, Springer, 2023, pp. 205–224.

[22] A. Gupta and C. Charan, "Analysis of Universal Asynchronous Receiver-Transmitter (UART)," in *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, IEEE, 2024, pp. 194–198.

[23] O. Standard, "MQTT Version 5.0," *Retrieved June*, vol. 22, no. 2020, p. 1435, 2019.

[24] T. Stapko, "CHAPTER 2 - Network Communications Protocols and Built-in Security," in *Practical Embedded Security*, T. Stapko, Ed., Burlington: Newnes, 2008, pp. 23–48, ISBN: 978-0-7506-8215-2. DOI: https://doi.org/10.1016/B978-075068215-2.50003-3.

[25] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.", 2012.

[26] A. G. M. Standard *et al.*, "Protocol Specification," *Network Programming, SunOS*, vol. 4, pp. 1–69, 2006.

[27] Supabase, "Build in a weekend. Scale to millions.," [Online]. Available: https://nodered.org/ (visited on 01/14/2025).

[28] M. Masse, *REST API design rulebook*. " O'Reilly Media, Inc.", 2011.

 ISBN: 978-1-68558-236-4

[29] R. Boyd, *Getting started with OAuth 2.0.* " O'Reilly Media, Inc.", 2012.
[30] J. W. Applebaum, C. W. Peek, and B. A. Zsembik, "Examining US pet ownership using the General Social Survey," *The Social Science Journal*, vol. 60, no. 1, pp. 110–119, 2023.
[31] N. R. Gee and M. K. Mueller, "A systematic review of research on pet ownership and animal interactions among older adults," *Anthrozoös*, vol. 32, no. 2, pp. 183–207, 2019.
[32] S. Holay, S. Wankhede, N. Gaikwad, and S. Tandekar, "Understanding When to Use SSR and CSR in Modern Web Development," *Journal of Technical Education*, p. 277, 2024.
[33] K. Vallamsetla, "The Impact of Server-Side Rendering on UI Performance and SEO," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 10, no. 5, pp. 795–804, 2024.
[34] H. E. G. Corporation, "Technical Data MQ-7 Gas Sensor," 2004, [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf (visited on 01/14/2025).
[35] ArduinoModules.info, "KY-037 High Sensitivity Sound Detection Module," [Online]. Available: https://arduinomodules.info/ky-037-high-sensitivity-sound-detection-module/ (visited on 01/14/2025).
[36] DFRobot, "Gravity: Analog Soil Moisture Sensor for Arduino and Raspberry Pi (SKU: SEN0114)," [Online]. Available: https://www.dfrobot.com/product-599.html (visited on 01/14/2025).
[37] DFRobot, "DFRobot Capacitive Touch Sensor (SKU: DFR0030)," [Online]. Available: https://wiki.dfrobot.com/DFRobot%5C_Capacitive%5C_Touch%5C_Sensor%5C_SKU%5C_DFR0030 (visited on 01/14/2025).