Architectural Tactics extraction from Autoware

Vitor Casadei, PhD Candidate, 2024

Resources: vcasadei/Autoware-Architectural-Tactics-extraction (github.com)

Introduction

AUTOWARE

Open-source software stack for self-driving vehicles, built on the Robot Operating System (ROS). It includes all of the necessary functions to drive an autonomous vehicles from localization and object detection to route planning and control, and was created with the aim of enabling as many individuals and organizations as possible to contribute to open innovations in autonomous driving technology.

autowarefoundation/autoware: Autoware - the world's leading open-source software project for autonomous driving (github.com)

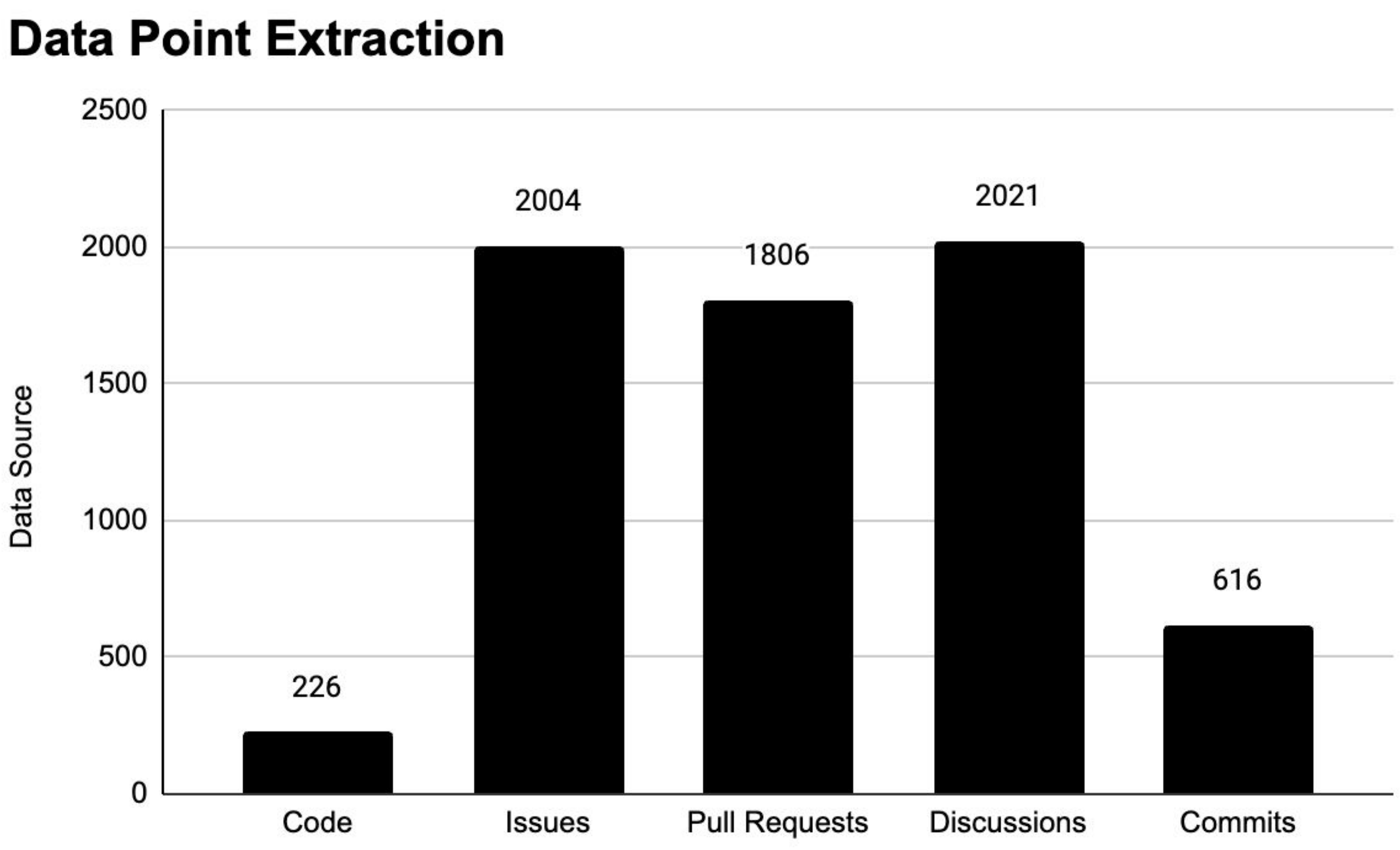
ARCHITECTURAL TACTICS

Architectural tactics are a key abstraction of software architecture, and support the systematic design and analysis of software architectures to satisfy quality attributes. [1]

Architectural Tactics are important as they have been defined by the experience of architects over the decades in a systematic set of architectural design decisions.

Architectural Tactics for Energy Efficiency are therefore architectural decisions with the goal of preserving energy, most of the times applicable on IoT or Mobile Devices.

[1] Gastón Márquez, Hernán Astudillo, Rick Kazman, Architectural tactics in software architecture: A systematic mapping study, Journal of Systems and Software, Volume 197, 2023, 111558, ISSN 0164-1212,



Data Point Extraction

The Data Extraction and Data Analysis process was inspired, but heavily simplified, on the paper [2].

To extract source code, linked documents, official documentation, Markdown files, issues, pull requests, commit messages and other information from the GitHub Repository.

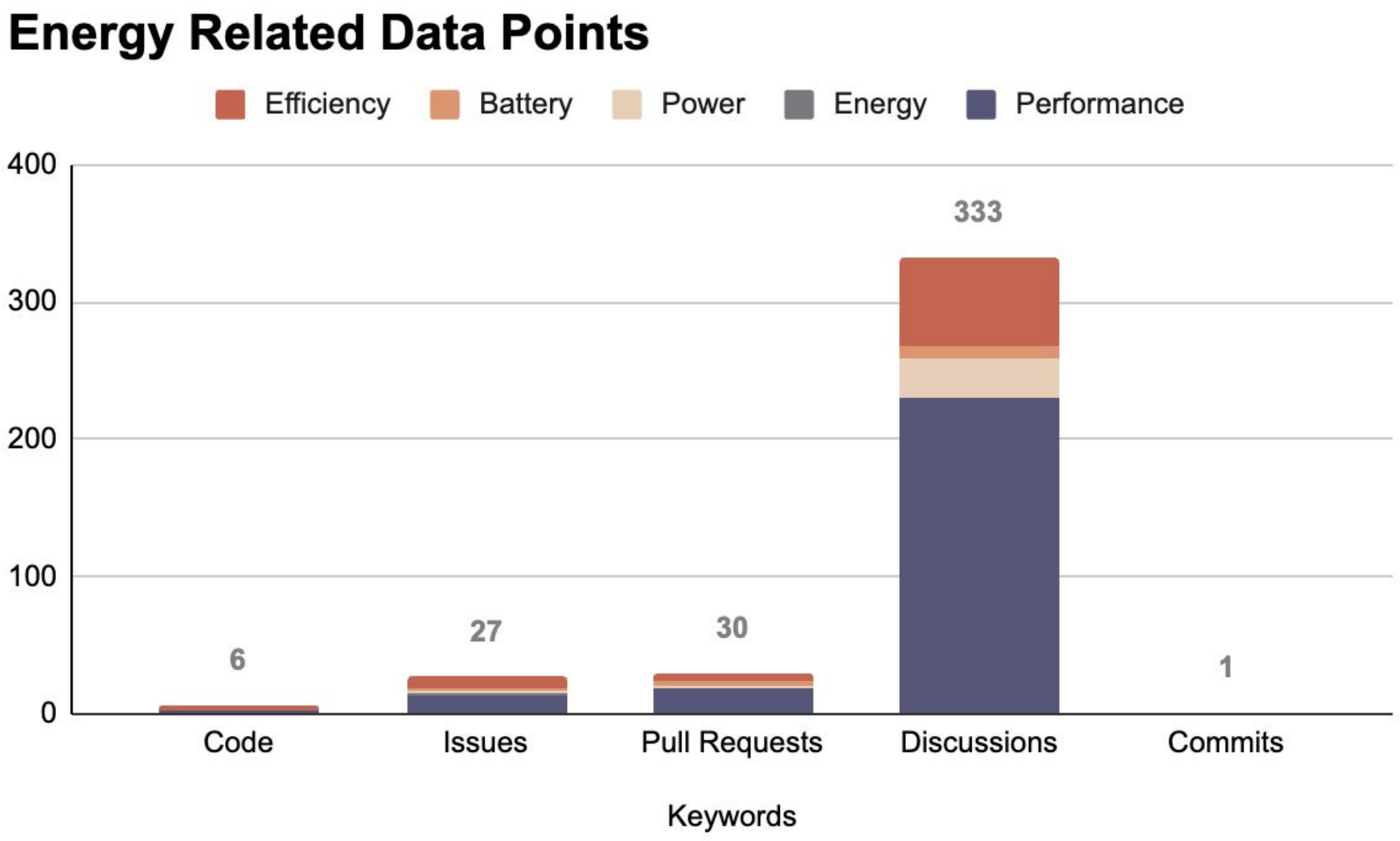
All Data Points were them stored using the software Notion as a Database. The choice of Notion was done because it was an easy to use tool and provided no setup and easy cataloging of information with filters and search functions.

At this point, a total of 6673 data points were collected.

[2] I. Malavolta, K. Chinnappan, S. Swanborn, G. A. Lewis and P. Lago, "Mining the ROS ecosystem for Green Architectural Tactics in Robotics and an Empirical Evaluation," 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), Madrid, Spain, 2021, pp. 300-311.

Identify Energy Related Data Points

In this process all Data Points were analyzed by the researcher with the goal of identifying those that mentioned energy-related topics. The GitHub Data Mining process was performed using keywords. This was done using GitHub's Code Search Tool [3].
The keywords used were "Efficiency", "Battery", "Power", "Energy" and "Performance" with minor variations of these words, such as "Efficient", for example.
At the end of this process, only 397 data points remained from the original 6673.



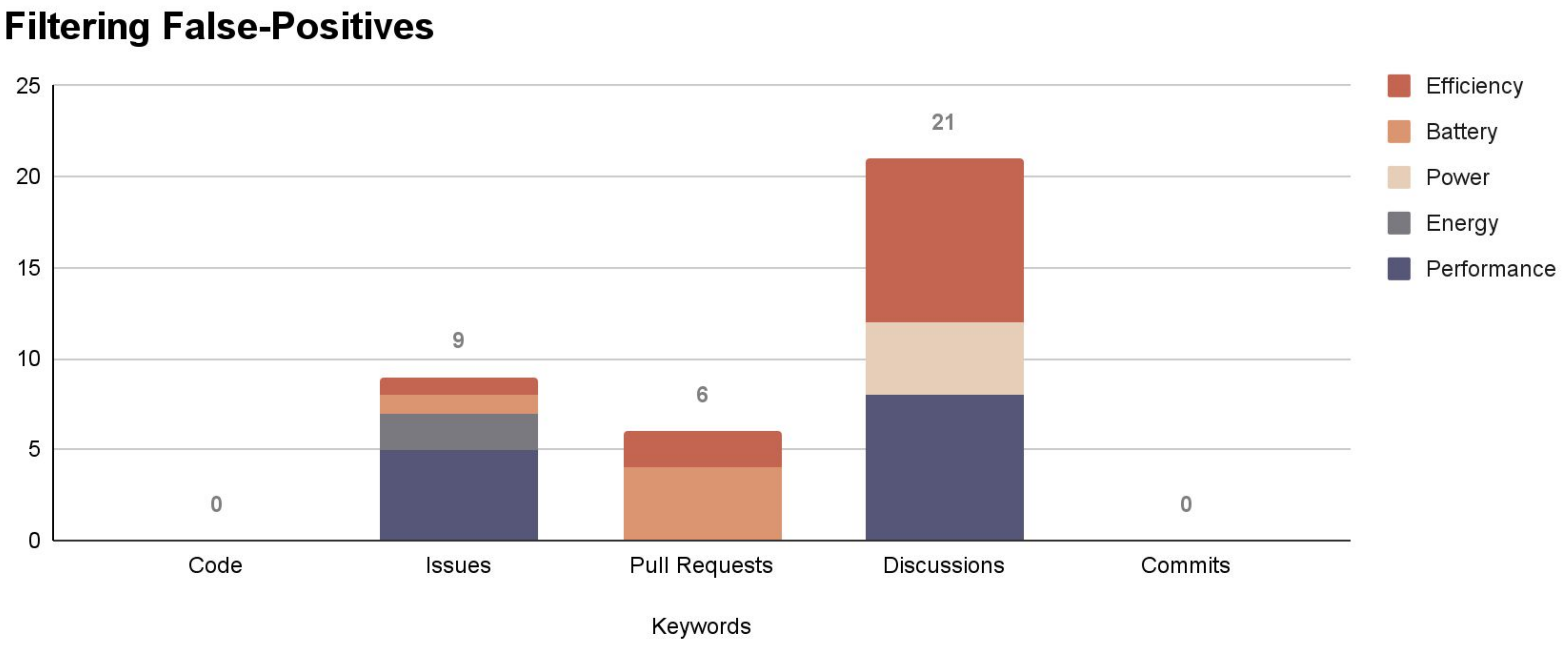
[3] GitHub Code Search

Filtering False-Positives

The next process was to perform an in-depth analysis of all 397 remaining data points in order to remove those that were False-Positives.

False-Positives are data points that referenced energy related topics, but were not related to energy saving or optimizations. For example, one discussion that was captured on the keyword battery was regarding the battery replacement in an autonomous kart project.

At the end of this process, only 36 data points remained.



Tactics for Energy Efficiency

Energy efficiency is, at its base, the optimal use of resources to perform an operation. That means that a scenario on energy efficiency is defined by the goal of preserve or manage energy while providing the necessary functionality. In this context, resources are computacional devices, such as CPU, memory, storage, network etc [4].

Architectural tactics for energy efficiency are defined in three categories by [4]: resource monitoring, resource allocation and resource adaptation. These categories and definitions are used in the subsequent analysis of the data points.

Monitor Resources

- **Metering**

"collecting data about the energy consumption of computational resources via a sensor infrastructure, in near real time"

- **Static Classification**

"estimate energy consumption by cataloging the computing resources used and their known energy characteristics"

- **Dynamic Classification**

"estimate energy consumption based on knowledge of transient conditions such as workload"

Allocate Resources

- **Reduce Usage**

"Usage can be reduced at the device level by device specific activities"

- **Discovery**

"discovery service matches service requests (from clients) with service providers"

- **Schedule Resources**

"allocation of tasks to computational resources"

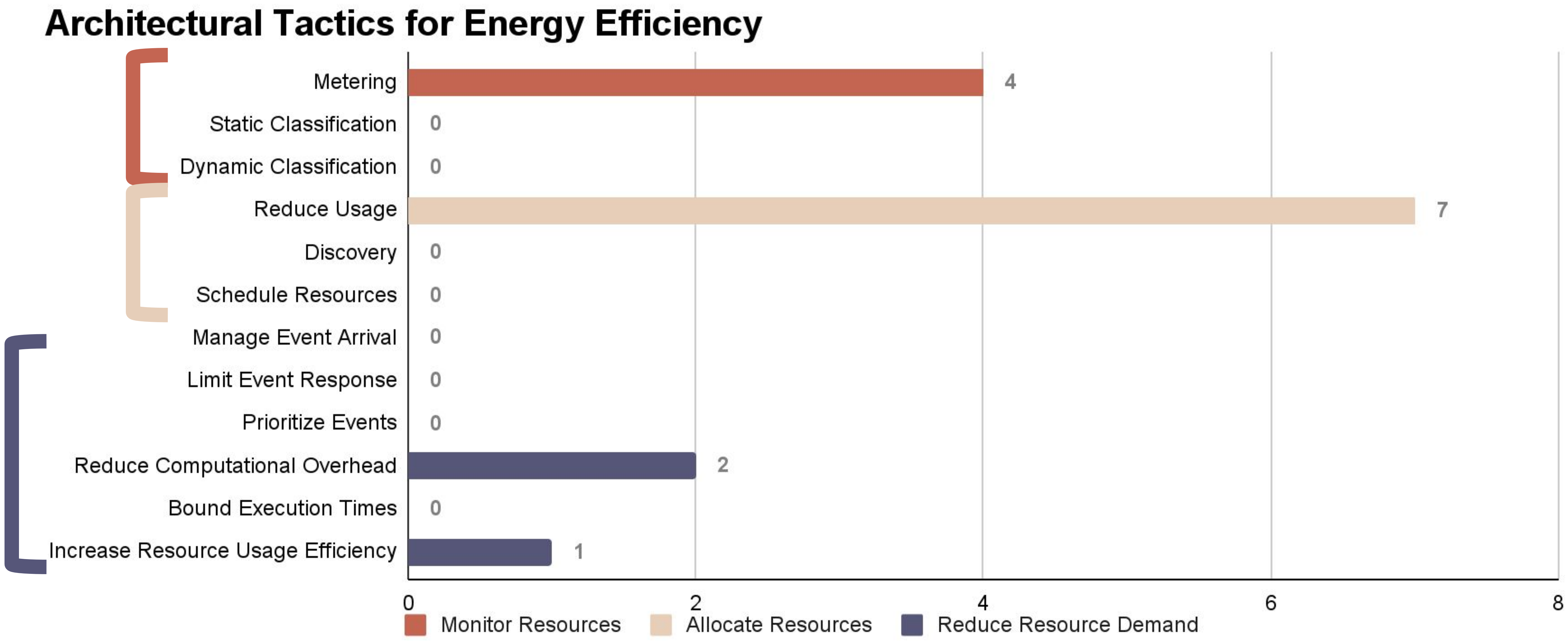
Reduce Resource Demand

- **Manage Event Arrival**
- **Limit Event Response**
- **Reduce Computational Overhead**
- **Prioritize Events**
- **Bound Execution Times**
- **Increase resource usage efficiency**

[4] BASS, L.; CLEMENTS, P.; KAZMAN, R. Software Architecture in Practice. Fourth Edition ed. [s.l.] Pearson Addison-Wesley, 2022. p. 140–154

Extract Tactics for Energy Efficiency

In this final step the 36 data points remaining are thoroughly analysed in order to identify those that discuss approaches of Architectural Tactics for Energy Efficiency. Based on the classification and categorization of [4], a **total of 14 instances** of such discussions were collected, with 4 regarding the Monitoring of Resources, 7 the Allocation of Resources and 3 the Reduction of Resource Demand.



[4] BASS, L.; CLEMENTS, P.; KAZMAN, R. Software Architecture in Practice. Fourth Edition ed. [s.l.] Pearson Addison-Wesley, 2022. p. 140–154

Tactics on **Resource Monitoring**

Metering

- Create api for vehicle information · Issue #3226 · autowarefoundation/autoware (github.com)
    - Discusses the addition of Energy Level
    - Energy levels ... provided as a percentage in vehicle status API. This API provides the following information, and the application can estimate the remaining operating time with vehicle status.
        - energy type (e.g. battery, fuel)
        - max value and unit (e.g. 10 [J], 10 [Wh], 10 [L])
        - approximate operation time
        - approximate driving distance
- Create api for vehicle status · Issue #3232 · autowarefoundation/autoware (github.com)
    - Similarly, discusses the addition of Energy monitoring.
- Add battery charging state by s-azumi · Pull Request #2256 · autowarefoundation/autoware (github.com)
    - Discusses the addition of a Battery Charging State into the Vehicle state machine for decision making.
- Support full functions of g30esli_interface by aohsato · Pull Request #2139 · autowarefoundation/autoware (github.com)
    - Discusses message communication with battery metrics, such as:
        - Battery Charge
        - Battery Voltage
        - Battery Amperage

Tactics on **Allocation of Resources**

Reduce Usage

- Proposal for splitting a single container image into multiple container images · autowarefoundation · Discussion #4661 (github.com)
    - Discusses creating containers for specific modules/packages to make easier to coordinate the computational resources of each Autoware component.
- Perception Containerization · Issue #4646 · autowarefoundation/autoware (github.com)
    - Discusses containerization with a goal of resource optimization and dynamic resource allocation.
- Split the `planning`/`control` packages into a separate container · Issue #5081 · autowarefoundation/autoware (github.com)
- Split the `mapping`/`localization` packages into a separate container · Issue #5080 · autowarefoundation/autoware (github.com)
- Split the `sensing`/`perception` packages into a separate container · Issue #5079 · autowarefoundation/autoware (github.com)
    - Implements on the above discussions of splitting packages for dynamic resource allocation and optimization.
- Add independent camera module to Autoware · autowarefoundation · Discussion #2646 (github.com)
    - Discusses the addition of an independent camera module as it "usually needs more computing power, memory, and bandwidth than other modules".
        - So that the module could be easily decoupled when not in use.
- Autoware Transform Listener - performance improvement · Issue #5385 · autowarefoundation/autoware (github.com)
    - Discusses a dynamic allocation of resources to deal with different messages with the goal of "Reducing CPU load by getting rid of unnecessary TF listeners in ROS nodes graph."
    - This discussion even has graphs of a CPU load test with multiple implementations.

Tactics on **Reducing Resource Demand**

Reduce Computational Overhead

- [Proposal] An efficient way to handle topic messages by reducing invocations of subscription callback functions · autowarefoundation · Discussion #4612 (github.com)
  - Discusses a better way of handling callback messages in order to reduce computational overhead on the CPU from many unnecessary and repeated messages with a timer for receiving and possibly ignoring similar (non-critical) messages.
- Software WG Meeting 2024/04/23 · autowarefoundation · Discussion #4653 (github.com)
  - The same as the one above.

Increase Resource Usage Efficiency

- Put every node into a single component container? · autowarefoundation · Discussion #3818 (github.com)
  - Discusses splitting each node in a separate container with the goal of improving Memory Usage.
    - "Running all nodes in a single process can sometimes result in better memory usage, as there's only one instance of the ROS2 runtime and potentially fewer duplicated resources."

Discussion

Optimization/Energy Efficiency

There is a fine line between code or architecture optimization for better performance or lower resource usage and actual energy savings from energy efficient tactics.

Choosing Keywords

Some keywords have multiple meanings, resulting on many False-Positives and others are too specific and are not used in discussions.

Evidence of Tactics

In many cases the evidence of architectural tactics is not clear in a discussion, PR, MR, documentation or code alone, but is linked to another data point that explains and corroborates it.

Interpretation

The interpretation of a single researcher is biased and as it's not common for developers to directly approach energy efficiency, the researcher needs to constantly interpret data points and try to relate with the energy topic.

Real World Effectiveness

Most discussions and data points do not provide evidence of actual effectiveness that an architectural tactic is going to have the desired result (of lower resource usage for example). Therefore, it's difficult to analyze if an action of architectural decision in fact results on energy efficiency or energy savings.

Thank You

Resources: vcasadei/Autoware-Architectural-Tactics-extraction (github.com)