2014•2015
FACULTEIT WETENSCHAPPEN
master in de informatica

Masterproef

RuleMaDrone: A Web-Interface to Visualise Space Usage Rules for Drones

Promotor :
Prof. Dr. Johannes Schöning

Aäron Trippaers

Scriptie ingediend tot het behalen van de graad van master in de informatica

universiteit hasselt | Maastricht University

Abstract

During the last years we have seen a rise of consumer and professional drones. There are different companies manufacturing relatively cheap consumer drones for everyone. These models are quickly gaining popularity. The professional world is also looking into drones. Logistics companies are testing drone delivery services and construction companies are experimenting in maintenance of higher structures with drones.

The rise of professional and consumer drone usage also resulted in safety issues. Every week we see news reports about drones flying too close to commercial airplanes around airports or above nuclear power plants. If an electronic device such as a drone crashes into an airplane or a nuclear power plant, the results are catastrophic. Solving these problems makes integrating drones controlled by private individuals or organizations a challenge for the coming years. Governments are in the process of creating, proposing and modifying laws to solve these safety problems. But the rapid changing of laws and regulations does not reach drone operators. This creates a dangerous situation where drone operators are not familiar with the rules and regulations to which they have to comply with by law.

RuleMaDrone, an application developed within this thesis, is presented as a solution to communicate the rules and regulations to drone operators. To provide the solution a framework for drone safety was designed which consists of the rules and regulations, the drone properties and the environmental factors. RuleMaDrone is developed with this framework and thus will provide drone operators with an application which they can use to find a safe and legal fly zone. RuleMaDrone uses a natural language based interface to add or edit the rules. Rules proposed via this interface are generated automatically. The web application is

flexible, making it possible to use it for other space usage rules as well.

Dutch Summary

RuleMaDrone: Veilig Vliegen Begint Hier!

Dankzij de vooruitgang in de technologie zijn geavanceerde drones de laatste jaren alsmaar betaalbaarder geworden. Je hebt al een heel simpele drone voor ongeveer 30 euro. Voor ongeveer duizend euro kan je al een heel erg goede drone kopen met de laatste snufjes: een goede camera, de laatste beveiligingsopties en een controller die live beelden doorkrijgt van je drone. De toepassingen van deze vliegende robots zijn eindeloos. Amazon wil ze gebruiken om snelle pakketlevering uit te voeren, Matternet gebruikt drones om afgelegen gebieden van medicijnen en voedsel te voorzien en vele mensen kopen een drone om fantastische foto's en filmpjes te maken.

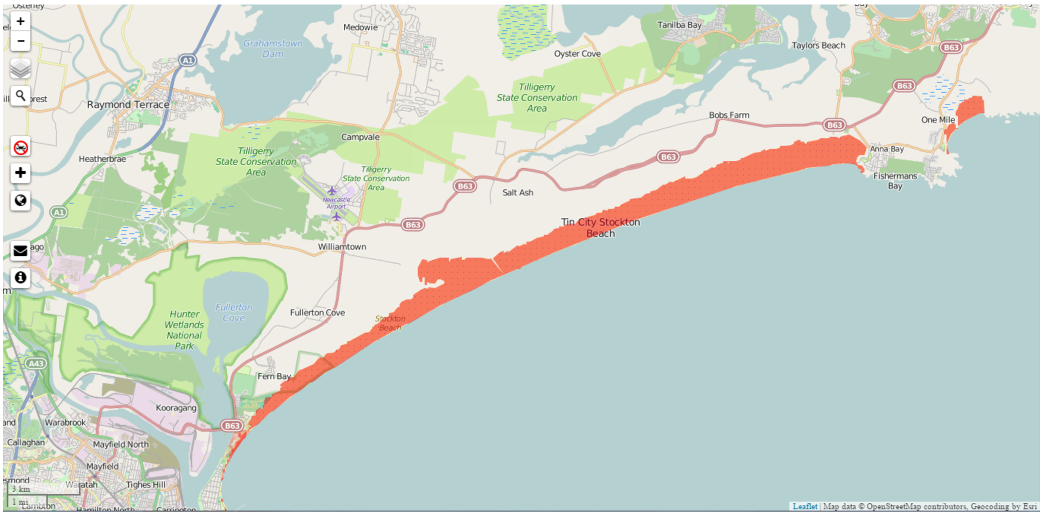
Het laatste jaar kwamen deze populaire kerstcadeaus jammer genoeg vaak op een negatieve manier in het nieuws. Zo werden er drones gezien boven verschillende Franse kerncentrales waar het verboden is om te vliegen. Men wil vermijden dat iemand met slechte bedoelingen schade kan aanrichten aan de kerncentrales. Op verschillende plaatsen ter wereld werden er ook drones gesignaleerd rond luchthavens. Dit is eveneens verboden. Een drone kan passagiersvliegtuigen die landen of opstijgen hinderen of beschadigen. Deze incidenten waren de aanleiding voor vele mensen om drones als iets slecht te gaan beschouwen zoals ook de volgende quote aangeeft:

"When I started working in this field, the word 'drone' wasn't used at all!" Sergei Lupashin says a little plaintively. "We called them UAVs, unmanned aerial vehicles, and they were clearly flying robots. Nowadays it's all a bit more mixed up; you have to take extra steps to convince

people that what you're doing is flying a cell phone and not a miniaturized Predator."[1]

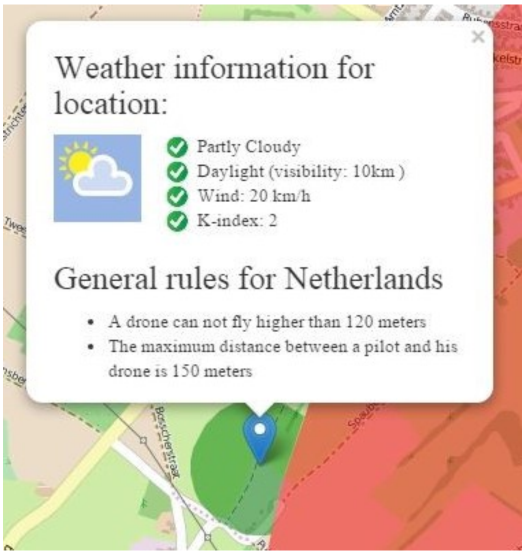
Om deze perceptie te veranderen moeten we een manier vinden om te voorkomen dat mensen op deze verboden locaties gaan vliegen. Een van de redenen dat mensen drones vliegen over deze verboden gebieden is het gebrek aan kennis over de regelgeving rond drone-gebruik. In een door ons uitgevoerde survey waarin 203 personen deelnamen gaf ongeveer zeventig percent aan geen kennis te hebben van de wetgeving rond drones in hun land. Om hier verandering in te brengen besloten wij een applicatie te bouwen die drone-piloten op een simpele manier weergeeft waar ze veilig en legaal kunnen vliegen.

Net als Wikipedia niet zelf zijn artikelen schrijft, gaan wij niet zelf alle regels verzamelen. Op onze website kunnen mensen die de regelgeving rond drones in een bepaald land kennen deze delen met andere drone-piloten. De wetgeving wordt ingeven door de regels om te zetten naar gestructureerde zinnen. In onze interface kan iemand bijvoorbeeld de zin "Do not fly your drone over beaches in Australia" bouwen. De webmaster van de site krijgt dan deze zin samen met een link naar een website die de regel bevat te zien op zijn dashboard en kan zo de regel controleren en toevoegen aan de website. Het enige wat de webmaster daarvoor moet doen is de regel als correct aanduiden, de rest doet de computer. Enkele ogenblikken later kan je dan op de website zien dat



[1] http://ideas.ted.com/take-a-flying-drone-for-a-walk/comment-page-1/

alle stranden in Australië rood gekleurd zijn zoals op de afbeelding hierboven. Als de regel daarna wordt aangepast zodat men niet alleen over het strand maar ook 100 meter rondom het strand niet met een drone mag vliegen kan een gebruiker dit gemakkelijk aangeven met dezelfde interface.



Naast het weergeven van de locaties waar het verboden is om een drone te vliegen, bespaart RuleMaDrone een drone-piloot ook extra opzoekwerk. Wanneer hij een locatie gekozen heeft waar hij wil gaan vliegen kan hij op onze website zijn drone aanduiden in een lijstje en de blauwe pin op zijn locatie naar keuze plaatsen. RuleMaDrone berekent dan voor hem of de omstandigheden om met een drone te gaan vliegen goed zijn. Zoals je op de afbeelding rechts kan zien zijn de omstandigheden goed om een drone te vliegen op deze locatie. Er is daglicht met een goede zichtbaarheid, de wind waait niet te hard voor de drone die hij gekozen heeft en het regent er niet. Op de kaart zelf ziet de drone-piloot dan een groene cirkel verschijnen die zijn fly zone genoemd wordt. Deze groene zone is de zone waarin hij mag vliegen. De grootte van deze zone hangt af van het bereik van de drone, de wetgeving in het land waar hij wilt gaan vliegen en de zichtbaarheid. Bij een dikke mist bijvoorbeeld wordt de zone verkleind. In dit geval heeft de zone een straal van 150 meter omdat de regelgeving in Nederland voorschrijft dat de afstand tussen de piloot en zijn drone niet groter mag zijn dan dat. De piloot kan zijn fly zone ook in 3D bekijken binnen onze applicatie zoals je hieronder kunt zien.

Als de piloot een locatie zou kiezen waar het weer niet goed genoeg is dan zou de tekstballon er uitzien zoals op de afbeelding aan de linkerkant. Hierop kun je zien dat de wind te hard waait om met de door jou gekozen drone te vliegen. Het is ook nacht wat het ook onveilig maakt. Verder regent het ook nog wat slecht is voor je drone. De site werd in de eerste weken dat hij online stond al meer dan duizend keer bezocht door bezoekers over de hele wereld. Er zijn genoeg redenen om nu als de bliksem naar www.rulemadrone.org te surfen maar we sommen er nog even enkele voor je op.

- Je bent een drone-piloot en je wil je volgende vlucht plannen.
- Je kent de regelgeving in een land en wil andere mensen helpen door deze in te geven.
- Je wilt kijken of je met je volgende drone op je gedroomde locatie mag vliegen.
- Je bent razend enthousiast over dit project en wilt er meer over weten.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

The Google Trends graph for the term ’drone’ in Fig. 1.1 captures the rise in popularity of drones since 2010. The letters on the graph refer to popular articles at that time. All but one article are about military events: ’Israel shoots down Gaza drone’, ’Drone kills Al-Qaeda suspects in Yemen’ and ’Iran airs footage from downed US drone’. This is an expected result since the vast majority of the budget spent on drones is for military drones as can be seen in Fig. 1.2. In this thesis however, we are interested in two other drone sectors that are steadily rising: consumer and professional drones.

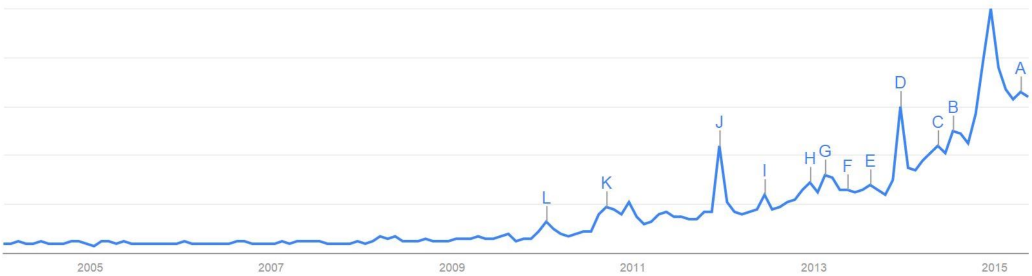


Figure 1.1: Rise in popularity of the term ’drone’ in Google Trends

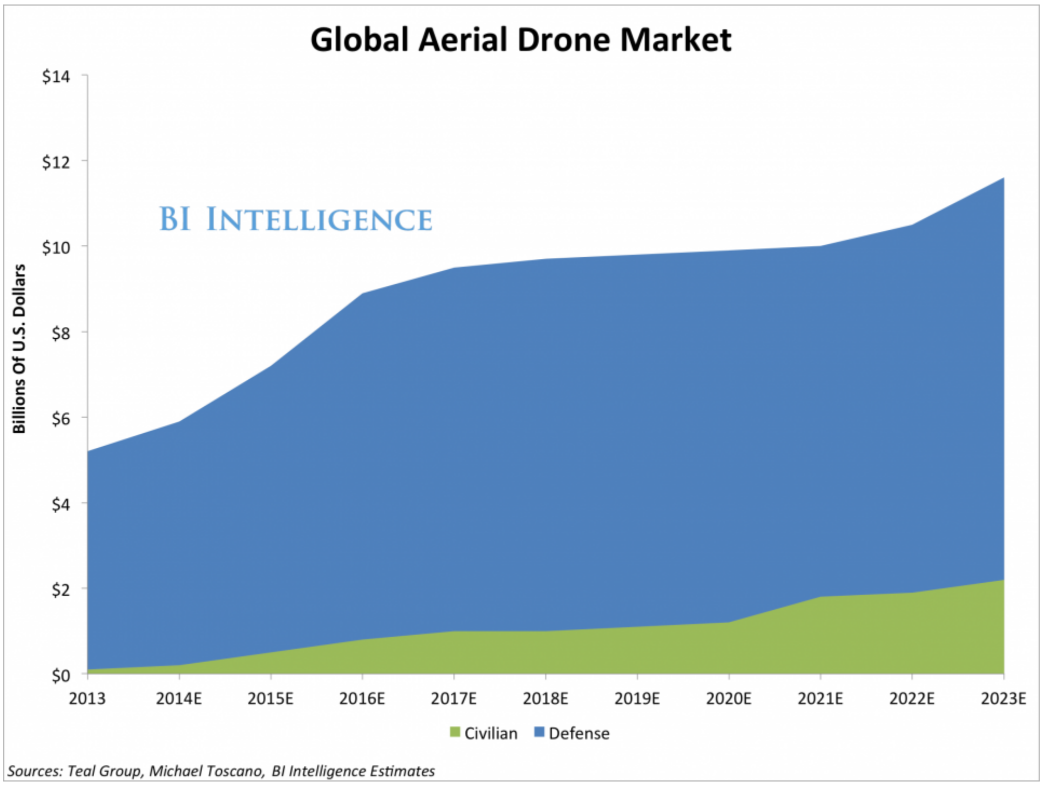


Figure 1.2: A comparison on predicted budget spent on drones for the next years between military and civilian drones

1.1 History of Drone Usage

The idea of unmanned flight was pioneered on August 22, 1849.[1] In order to attack Venice, the Austrians launched hot air balloons filled with explosives. These balloons laid the foundations of unmanned flight. When manned flight was invented in 1906, the concept of unmanned flight was still strong enough that research in it started immediately afterwards. It was not until World War II before military drones were used regularly.

In the 1930's Reginald Denny produced the first radio controlled aircraft for the consumer market. Despite the fact that these drones were popular throughout the decades afterwards, they were only affordable and controllable by flight enthusiasts. Recent developments in technology made drones cheaper, smaller and more affordable for everyone. In this same period that the purchase price of drones decreased, the professional world also started to research the possibilities and a whole new range of applications came into existence. One of the most well known examples are the delivery drones of Amazon.

1.2 Professional Drones

To prove that consumer and professional drones are rising in popularity we focus on two different peaks on the Google Trends graph in Fig. 1.1: the highest peak at the end of 2014 and the peak with the letter D. Article D has the title: 'Amazon unveils drone delivery plan'. This article explains how Amazon plans to innovate their delivery services with drones. Amazon's delivery drones are only one example in the professional drone market. At the EUKA drone convention in Genk many examples of how drones could change the professional world were shown. Most of these examples were already in their testing phase or even being used. Mr. Geert Nijst, a so called 'Drone Evangelist' working at Skyeye, elaborated on how many drones will be used in the professional world within a few years.

[1] https://understandingempire.wordpress.com/2-0-a-brief-history-of-u-s-drones/

Construction Drones are easy, fast, cheap and reliable to perform regular inspection on wind turbines, bridges and transmission towers. This is only a very limited list of structures where maintenance using drones is preferable to current methods. A lot of drones will be required world wide if we start using them for maintenance of these structures. According to OpenStreetMap there are around 13 million of these structures (table 1.1) that will be subjected to yearly inspection using drones one day.

Object	Tagged
Bridges	2 481 523
Tranmission towers	10 477 009
Wind turbines	20 981

Table 1.1: Amount of structures tagged in OpenStreetMap that benefit from yearly inspection by drones.

Logistics Amazon is testing drones to deliver packets. According to Jef Bezos 86 percent of the packets are under 5 pounds, which is easy to carry for a drone.[2] Drones have the advantages of being cheap and eco friendly in comparison to road transportation. On Ivy Bussiness review Prashob Menon makes an estimate that Amazon would need thousands of drones to deliver only 17 percent of the packets in North-America.[3] In Hong et al. (1) they propose a solution for one of the major problems with drone delivery: how to work with the limited amount of battery time.

These are only two examples where drones will be an important part of further innovation. A huge increase in the amount of drones flying around will be inevitable. There are also a lot of other sectors where drones are tested and even already being used: security, agriculture, humanitarian help, police and fire fighters. In Elzweig (2) more information about the possible use cases and the current problems of commercial drones can be found.

[2] http://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/

[3] http://iveybusinessreview.ca/blogs/pmenonhba2010/2013/12/11/how-many-drones-does-amazon-need/

1.3 Consumer Drones

The highest peak on the Google Trends graph (Fig. 1.1) is on December 25, 2014 but there is no article where this part of the graph refers to. One can easily deduct that drones were a popular Christmas present by looking at the date and doing a Google search for that period. A graph with drone sales on eBay per week (Fig. 1.3) shows that there were indeed a lot more drone sold in December 2014 than during the rest of the year. Around 7500 drones per week were sold in that period on eBay alone.

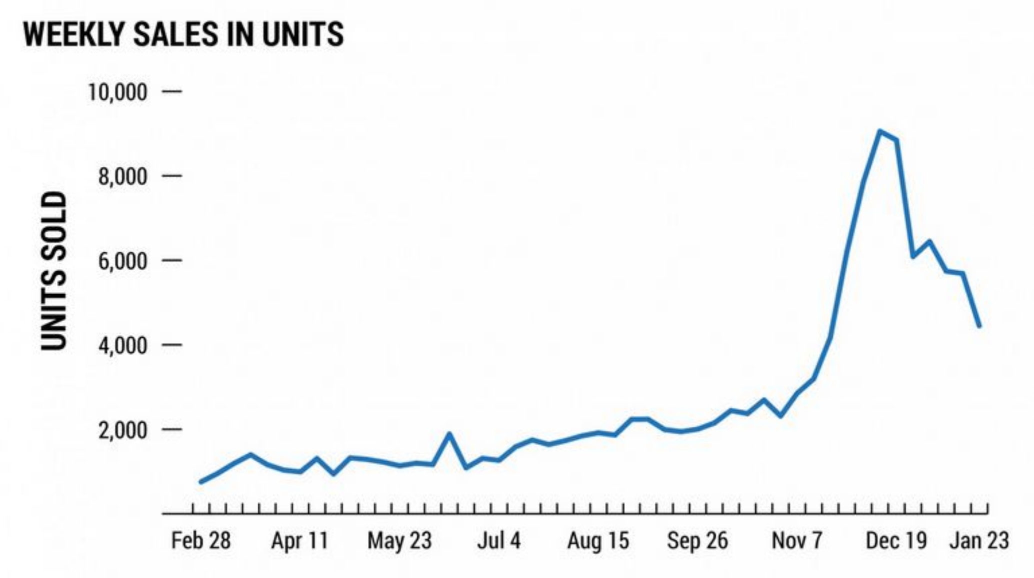


Figure 1.3: Weekly amount of drones bought on eBay plotted over time

1.4 Safety Problems

Drones are increasing in popularity, so a lot more drones are flying around. This resulted in an increase of drone incidents. Some news reports of drone incidents during last year are listed below.

- In the Netherlands a person was arrested for flying his drone over a firework show.[4]

[4] http://dronelife.com/2014/08/18/drone-pilot-arrested-netherlands/

- In the UK an aerial photographer was arrested while flying his drone over land for which he had permission of the land owner. This resulted in a dispute between him and the police.[5]
- Two men were flying a drone in New York and nearly collided with a police helicopter. These two men were also temporarily arrested.[6]

These issues are the reason why parallel with the term 'drone' the term 'drone safety' also started growing, see image Fig. 1.4. The term 'drone safety' was mostly non-existent until it started rising to its highest peak in December 2014. These safety problems were a cause for most governments to limit or even temporarily ban consumer and/or professional drone usage. Most people were, however, often unaware of these new regulations. In Roos (3) it is stated that deficiencies in knowledge can cause a drone operator to unwittingly violate laws and regulations. In Clarke and Moses (4) the relation between the laws and regulations and public safety is further explained. The safety issues with drones are further explained in Carr (5). When laws and regulations are broken, dangerous situations arise. Therefore it is important to provide a platform where drone operators are able to search for legal and safe fly zones. It is of major importance for such a platform to be adaptable to the changing of rules and regulations.

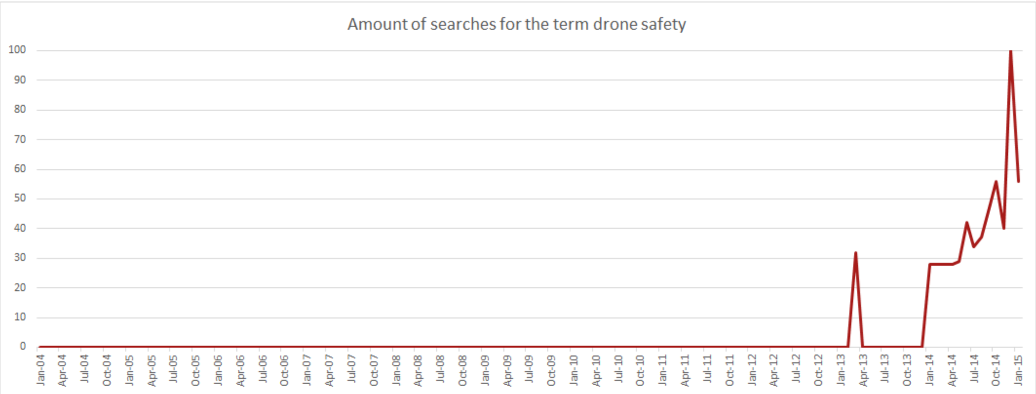


Figure 1.4: Amount of searches for 'drone safety'

[5] http://www.theguardian.com/world/2014/dec/31/photojournalist-police-arrest-drone-complaints
[6] http://www.nydailynews.com/new-york/uptown/drone-hits-nypd-helicopter-2-men-arrested-article-1.1858159

1.5 Addressing the Safety Problems

The foundation of drone safety exists out of three intertwined components: laws and regulations, drone properties and environmental factors. Flying a drone safely requires these three elements to be checked. Our drone safety framework is visualised in Fig. 1.5. Each component is equally important to ensure a safe flight and all components are dependent on each other. In table 1.2 six questions that each drone operator must ask himself before flying are presented. Each question is categorized in one or more of the three components. The questions range from *Can my drone handle the wind?* to *Am I too close to an airport to operate a drone?*. In Fig. 1.6 the questions are linked to the components of the drone safety framework.

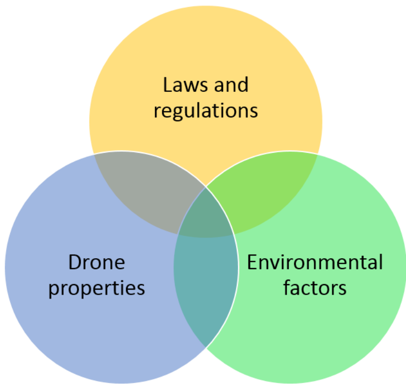


Figure 1.5: Framework of drone safety

Question
1) Am I too close to an airport to operate a drone?
2) Is it legal to fly a drone at night?
3) Do I have permission to operate drones heavier than 5kg
4) Can my drone handle the wind?
5) Is the visibility sufficient to fly a drone?
6) Is my battery charged?

Table 1.2: Questions a drone operator must ask himself before flying his drone

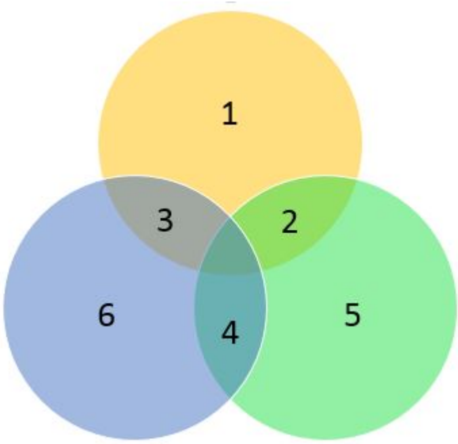


Figure 1.6: The questions linked to the drone safety framework

1.5.1 Laws and Regulations

An application visualising the laws and regulations must be global. It must be possible to integrate all the rules of all countries. To easily add these rules there must be a feature included to insert new rules or edit existing rules. When one is flying for entertainment the application must provide the user with an option to search for recommended places: these are remote controlled aircraft fields. When an application is able to include all these features, the regulation component is completed.

1.5.2 Drone Properties

As can be seen in the related work the drone properties component is the most neglected component in currently existing work. Not one rule mapping application makes a distinction between drones based on weight, controller range, GPS or the fact that a camera is mounted on it. Nevertheless there is for example a major difference between a toy drone of less than 1 kilogram and a professional camera mounted drone weighing 12 kilograms in terms of safety (6).

1.5.3 Environmental Factors

The weather is the most obvious environmental variable which will influence a drone. Wind will for example influence a drone's stability and mist will cause a bad visibility for the operator. Another factor that drone pilots must take into account is solar activity. High solar activity can cause erroneous GPS data. This can result into a fly away of the drone. Events taking place at your flight spot can also cause problems. In the United States one can not legally fly over crowded stadiums and flying through a firework show in the Netherlands is also prohibited.

1.6 Writing Style Conventions and Vocabulary Notes

- The word 'drone' always refers to a professional or consumer drone and not to a military drone.
- A drone operator is the person who controls the drone.
- Images are created by the author unless there is an URL of the image specified in the list of figures.

Chapter 2

Related Work

This chapter will situate this thesis in the current state of no-fly-zone applications for drones. First an overview is provided of the general research towards mapping space usage rules. The need for a new technique to collect space usage rules is explained and compared to an already existing approach. Thereafter some already existing tools regarding no-fly-zones will be discussed against our drone safety framework. This will provide the reader with a clear understanding of what our application will contribute.

2.1 Mapping Activity Restriction in General

In Schöning et al. (7) the lack of space usage rules in online mapping tools is discussed. A space usage rule (SUR) is defined as *SURs are a critical mechanism through which governments and other stakeholders (e.g. landowners) manage our interaction with our environment* (8). Examples of such space usage rules are no smoking, no fishing and no dogs.

Space usage rules are sparsely mapped on online mapping applications. To collect space usage rules for online mapping applications Samsonov et al. (8) proposed a computer vision approach. The application tags an OpenStreetMap feature based on a geotagged photo with a space usage rule sign. Which means that it has two preconditions:

- A geotagged picture of a space usage rule sign.
- The feature should already be included in OpenStreetMap.

The correct feature is selected with the computer vision algorithm and then tagged with the space usage rule.

This approach is efficient for space usage rules such as ’no smoking’ and ’no fishing’. But when this approach is used for implementing rules for drones the preconditions will cause multiple problems:

- No-drone-signs do rarely occur. Although some natural parks have a sign stating that drones are prohibited, these signs are mostly non-existent.
- Features that must be tagged do not always exist. When the rule *do not fly your drone in eight kilometres around a runway* is parsed the computer vision approach is only able to tag the runway since the polygon of the eight kilometre radius around it does not exist.

In this thesis these two drawbacks will be solved by using natural language to construct rules and combining the tagging of OpenStreetMap features with layer generation of each rule. Other work in visualising laws and regulation can be found in (9) and (10).

2.2 No-Fly-Zone Maps for Drones

However, there is a general lack of space usage rules in online mapping applications. It appears that there are some applications and maps regarding no-fly-zones.

2.2.1 DJI - Fly Safe

Major drone manufacturer DJI has a Fly Safe section on its website. A no-fly-zone map can be found in this section. DJI chose to categorize all commercial airports. They divided the airports into two categories:

Category A consists of large international airports such as Heathrow in London, Charles de Gaulle in Paris and John F. Kennedy International Airport in New York.

Category B Category B are smaller commercial airports such as Brussels South in Charleroi, Faro Airport and Eindhoven Airport.

Category A airports are displayed on the map with a no-fly-zone of 1.5 miles and another zone of 3.5 miles around it with certain height limits. Category B airports are displayed on the map with a no-fly-zone of 0.6 mile radius. Directly beneath the map there is a comprehensive list of all the airports included in the map with the type and the name of the airport.

DJI currently integrates this map in its drones, GPS data in the software of the drone will prevent a drone operator from flying a drone into a restricted area. The map is rather limited in comparison with the options implemented in the drone. The drone operator can limit the maximum height of the DJI Phantom and the maximum distance between him and the flying DJI Phantom.

2.2.2 Mapbox - Don't Fly Drones Here

In July 2014 Mapbox released a map visualising the drone regulations in the United States as a part of their project *Drones For Good*. Their map highlights areas where drones are prohibited. Mapbox divided their highlighted areas into four groups. The first three rules are highlighted with red stripes: major airports, military area's and natural parks. The fourth group are the temporal flight restrictions (NOTAMS) highlighted in yellow. More information about a highlighted area is shown when one hovers over such an area. This information is rather limited. When hovering over an airport the name of the airport is displayed next to an icon which expresses: *forbidden due to the proximity of an airport* (Fig. 2.2).

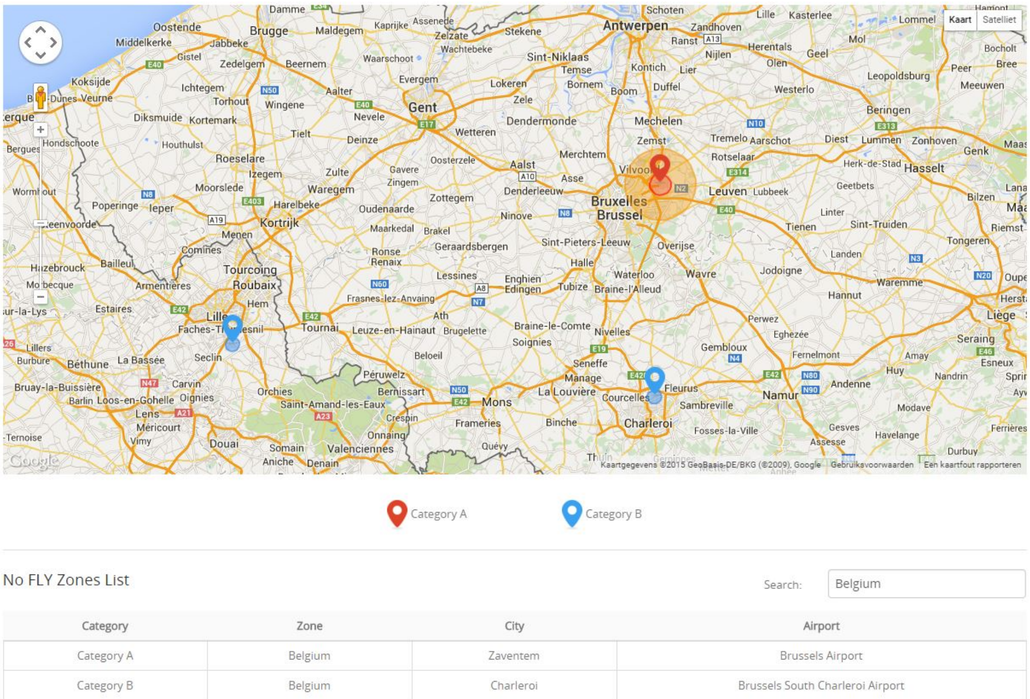


Figure 2.1: The Fly Safe map of DJI visualising the no fly zones around Belgian airports with a list of the airports beneath the map

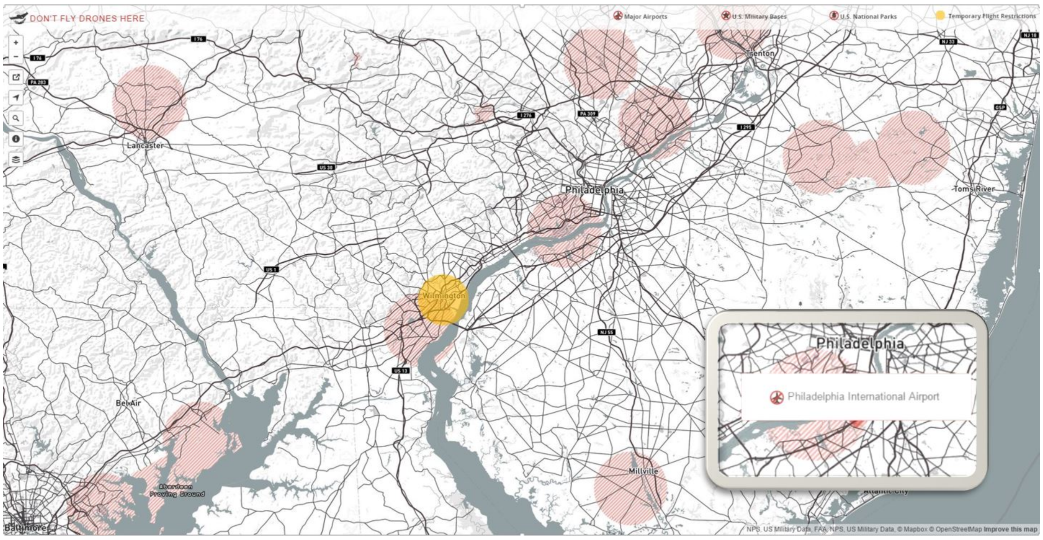


Figure 2.2: Don't Fly Drones Here by Mapbox, it is forbidden to fly drones in the red areas. The yellow areas are temporal flight restrictions. When hovering over one of these areas one can see the information displayed in the box

2.2.3 Hover

Hover is the only application in this list that has two out of three components of the framework integrated: environmental factors and laws and regulations. It shows the map provided by Mapbox discussed in section 2.2.2. So it automatically has all the features included in Mapbox. Next to this, it has a weather view in which the wind speed, wind direction and temperature are shown. Also the solar activity is displayed by the kp-index. Aside from these features the application also includes most recent drone news via RSS. Screenshots of the application can be seen in Fig. 2.3.

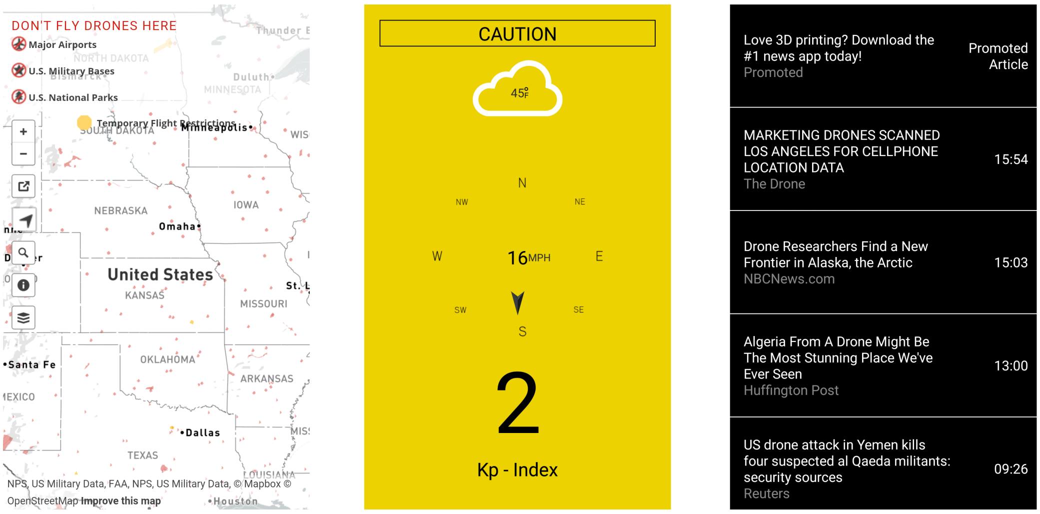


Figure 2.3: The map of mapbox integrated in Hover (left). The weather view of Hover with the wind speed, wind direction, KP-index and temperature at the current location (center) and an RSS feed of recent drone news (right)

2.2.4 RCFlyMaps

RCFlyMaps is an iPhone application that visualises the rules in the United States. The map works with satellite images overlayed by red zones where it is prohibited to fly a drone. They collected a lot of information from the FAA and other instances concerning the airspace of the United States so the rules would be as complete as possible. This was integrated with some social elements. Users are able to save their favourite flying spots. Places users liked and disliked are also visualised on the map with green and orange zones. Hobbyists fields are highlighted in blue. This provides a very broad range of options. The user interface did not allow to search the application very well but as far as we found, only the rules of the United States and Canada are visualised in this application. When one clicks on a coloured zone, information is shown about why it is forbidden to fly over this location and which user added this zone.

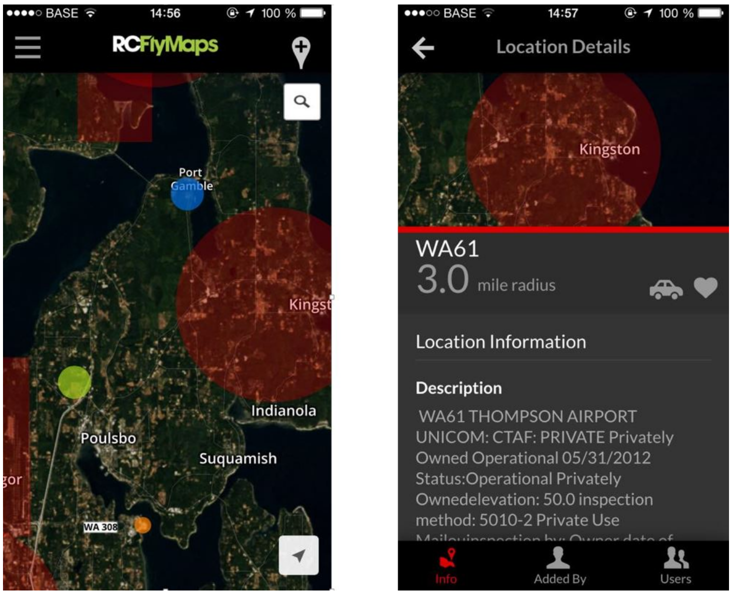


Figure 2.4: General overview and detail view of the RCFlyMaps application

2.2.5 RC Groups - RC Maps

RC Maps does not visualise any drone rules but cannot be excluded from this list because it has a similar functionality. The map created by RC Groups marks all the hobbyists fields for remote controlled cars, airplanes and boats. It also includes all the hobby shops. Including all the hobbyist fields where you are able to fly your drone is also a feature of our project in the recommended zones. Because flying on a hobbyists field is still one of the safest options for drone pilots.

Figure 2.5: RCMaps showing all the RC flying fields

2.2.6 Solar Activity Monitor

Solar Activity Monitor (SAM) is a very simple android application which issues a warning when there is too much solar activity to fly a drone. Solar activity can cause erroneous GPS data. Drones relying on GPS as a safety matter to fly back in case of a lost connection can fly in a wrong direction because of this GPS data. A more detailed explanation about the influence of solar activity can be found in section 3.2.2.

2.3 Advancements over Related Work

When comparing these freely available tools some shortages came forward. In chapter 1 we created a framework for drone safety. This framework will be used to discuss the shortages of these tools and the strengths of our own application. As can be seen in Fig. 2.7, none of the tools is complete according to our drone safety framework.

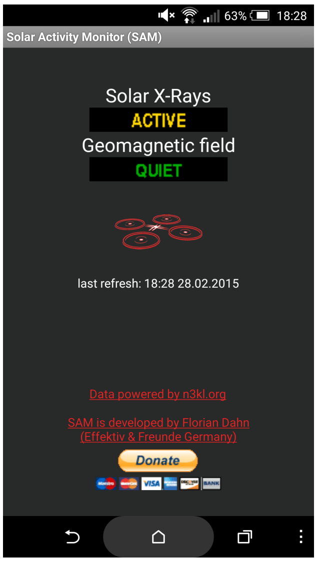


Figure 2.6: Screenshot of Solar Activity Monitor

2.3.1 Drone Properties

None of the tools discussed earlier has drone specific components. Which means that none of this tools integrate the rules concerning weight or the maximum range for a drone to fly. The pillar of drone properties is completely ignored in current work.

RuleMaDrone considers the drone-specific components. Weight, for example, appears as a parameter for general rules in RuleMaDrone. When it is forbidden in a country to fly drones heavier than ten kilograms, this will appear in the general rules. If the properties are provided the tool will provide feedback if the user can or cannot fly that type of drone at a specific location. When, for example, the wind speed is stronger than the drone can handle, a warning will be issued.

2.3.2 Environmental Factors

The environment is integrated into two of the discussed applications: Hover and SAM. SAM shows if it is dangerous to fly your drone due to solar activity but has no other features. Hover has the environment variables wind speed, wind direction, temperature and solar activity. It is the most complete application according to our framework because it implements two of the three pillars.

RuleMaDrone integrates these weather components in the map. Which makes it more flexible than Hover. In Hover the weather of the actual location is visualised but you cannot choose another location. In RuleMaDrone the weather of the chosen fly zone is extracted. Which makes it possible for a user to check the weather for his flying location in a distant area.

2.3.3 Laws and Regulations

To see how a tool complies to the laws and regulations component three factors are considered:

- Is the tool global?
- Does the tool integrate or try to integrate all rules?
- Does the tool have a function for recommended places?

In Fig. 2.7 it can be seen that most tools are not global. These tools all focus on the United States. The only tool that is global, Fly Safe from DJI, simply includes one rule.

It will take a long time before a tool has all the laws and regulations of all countries in one mapping application. RuleMaDrone will make this possible due to the open structure of the application. People are enabled to add regulations of any country to the map. Showing hobbyist fields or recommended flying zones to drone operators is vital. These fields are the safest places to fly a drone and sometimes even the only places to fly a drone.

Tool	Regulation	Environment	Drone	Free	Extra features
DJI - Fly Safe	Global One rule	✗	✗	✔	Drone integration
RCFlyMaps	United States All rules Hobbyist fields	✗	✗	✔	Social components
Mapbox - Don't fly drones here	United States All rules	✗	✗	✔	✗
RC Maps	Global Hobbyist fields	✗	✗	✔	✗
Solar Activity Monitor	✗	Solar activity	✗	✔	✗
Hover	United States All rules	✔	✗	✔	Drone news
RuleMaDrone	✔	✔	✔	✔	✗

Figure 2.7: Comparing the tools with our drone safety framework

Chapter 3

Concept of RuleMaDrone

RuleMaDrone is the first online mapping application for space usage rules according to the three pillars of drone safety: environmental factors, drone properties and laws and regulations. In this chapter we discuss for each pillar how it is separately integrated into RuleMaDrone.

3.1 Laws and Regulations

In the following section we discuss how the yellow pillar, laws and regulations, is integrated in RuleMaDrone. First the differentiation between spatial and general rules is made. Afterwards we discuss how these rules are visualised in RuleMaDrone. We conclude this section by introducing our new paradigm for adding and editing laws and regulations with natural language.

3.1.1 Rule Classification

Spatial rule A rule defined over a certain area due to the classification of that area (e.g. military area, theme park or beach) or the proximity of a certain object (e.g. building, road, vehicle or power line) in that area. These rules can be visualised on a two-dimensional map. Examples of spatial rules can be found in table 3.1.

General rule A rule defined over a whole country: independent of the classification of the area or the proximity of certain objects. Examples of general rules can be found in table 3.2.

Spatial rules
Do not smoke inside of buildings in Belgium
Dogs are forbidden in theme parks in the United Kingdom

Table 3.1: Examples of fictitious spatial rules.

General rules
Drugs are illegal in Germany
Do not drive a truck when it is raining in France

Table 3.2: Examples of fictitious general rules.

3.1.2 Rule Visualisation

It is important to correctly visualise the rules. Spatial rules are visualised as polygons on a two-dimensional map (Fig. 3.1). General rules are displayed when a user calculates the fly zone (Fig. 3.3). All rules are listed in the rule menu that is visible in Fig. 3.2. In this rule menu the user can switch countries to view the rules of another country. In this menu there are four actions that can be executed on a certain rule.

Turn rules on and off One can turn off a spatial rule layer. A rule that is turned off will not be visible and will not be used in the fly zone calculation. This will, for example, be useful when someone has received an extra permission for flying where it would normally be prohibited.

Changing the color To immediately recognize the different rule layers, one is able to change the color of a spatial rule. As shown in Fig. 3.2, it is easy to notice which polygon presents which rule. Assigning a color to each layer prevents the need for extra actions when a user wants to know which rule is presented by a polygon.

Get more information A user can get the link for more information on the rule. These are the same links that we use to verify the rules. See section 3.1.3.

Edit the rule One is able to edit the rule with the natural language interface described in section 3.1.3.

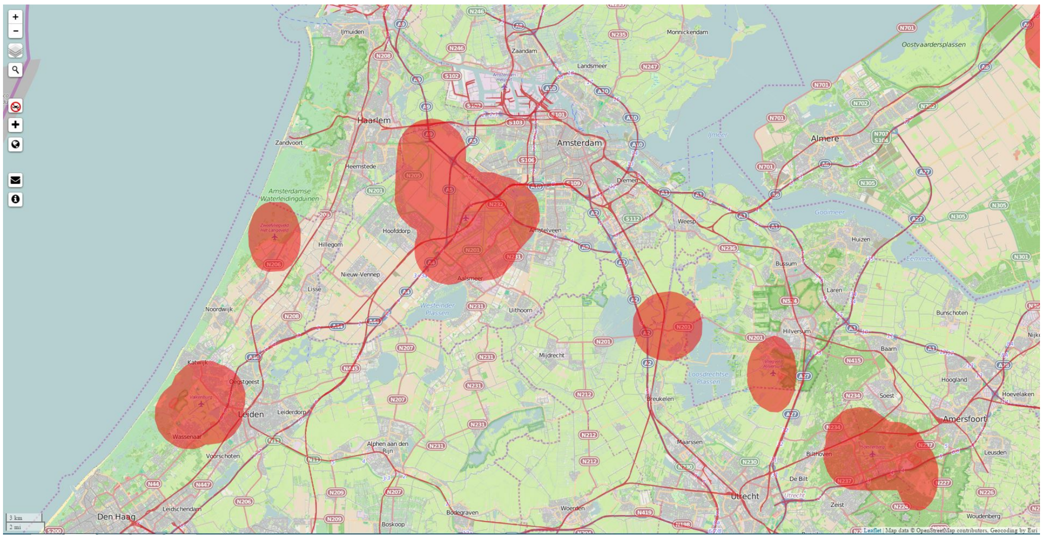


Figure 3.1: The map with the spatial rules visualised is the main element in RuleMaDrone

3.1.3 Adding and Editing Rules

A new paradigm of adding space usage rules to an online mapping platform is proposed in this thesis. Users are enabled to add rules to the map with a natural language-based interface. In order to remove ambiguity of natural language we provided the users with a predefined sentence where they can decide each part. The difference between general and spatial rules is very important in this case. The user must provide a link to prove that the rule he entered is valid.

General Rules: Parameter-Based Rule Adding

An important aspect of general rules is that they restrict drone flights due to some parameters, see table 3.3. General rules can thus easily be added by inserting the values at which drone flying is restricted or permitted. Fig. 3.4 shows how such a rule can be added. The user has to choose a value to restrict and fills in the empty parameter. In this example a drone operator would be limited to fly higher than 400

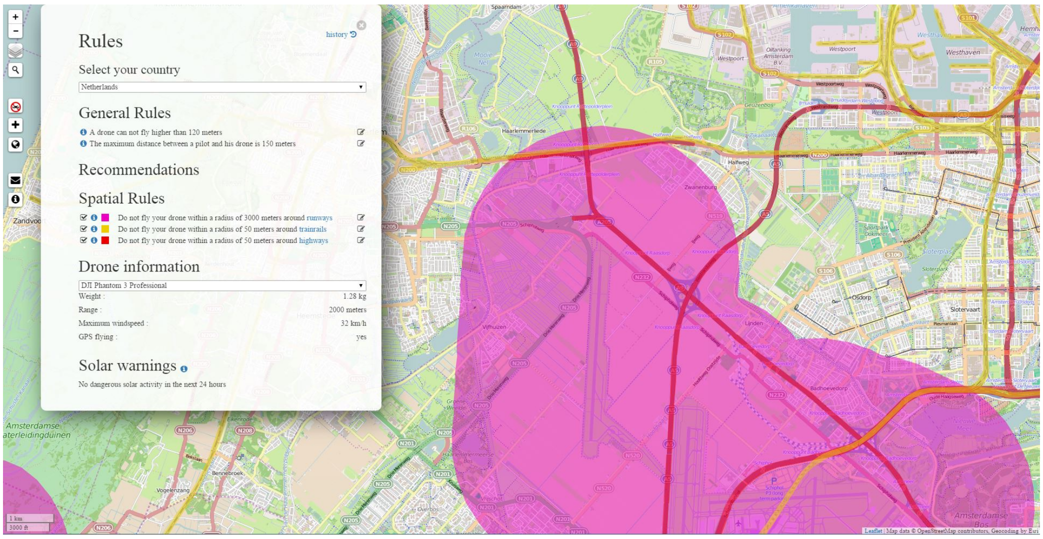


Figure 3.2: The rule menu is opened and the user has chosen to change the colors of the spatial rule layers

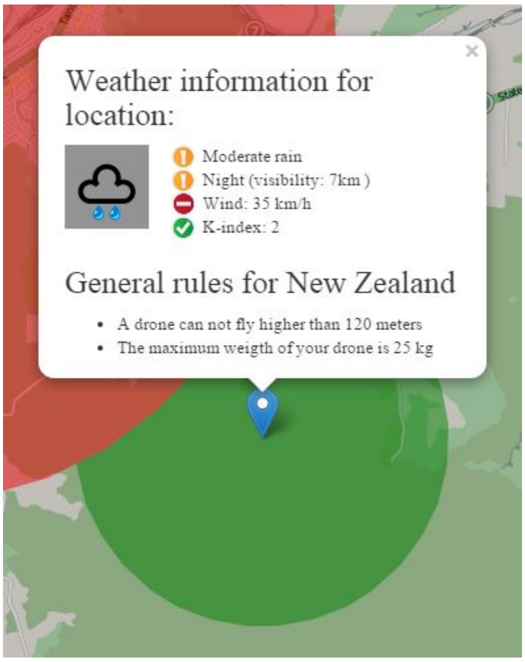


Figure 3.3: The fly zone pop-up with the general rules for New Zealand listed

meters. When a user wants to edit such a rule, he is provided with the same interface but can only change the parameter. In the rule *drones must fly lower than 60 meters above sea level in Germany* only the 60 can be edited into another number.

General rule	Parameter	Value
A drone cannot fly higher than 200 meters above sealevel	Height	<200
Only drones below 20kg can be flown legally	Weigth	<20
Drone flying is forbidden when there is mist	Weather description	mist

Table 3.3: General rules listed with their variable parameter.

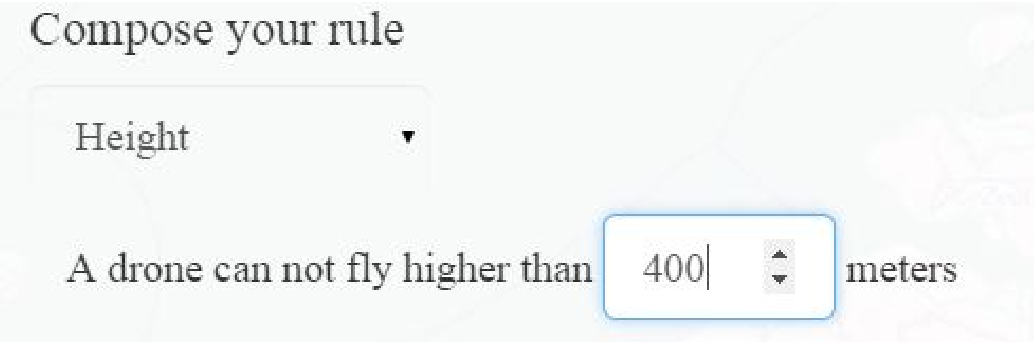


Figure 3.4: Parameter-based rule adding for general rules

Spatial Rules: Zone-Based Rule Adding

When adding a spatial rule the most important part is defining a certain zone in which such a rule applies. RuleMaDrone uses definitions composed by OpenStreetMap tags for defining such zones, different key-value pairs are mapped to a certain zone. Examples of these definitions can be found in table 3.4.

Once the zone is defined, it is very easy to create the whole rule. The user must choose between *do* or *do not* fly in that zone which will result in creating a recommendation or a restriction. If not only the zone is important but also a certain region around that zone the user can also define a radius. This is required for rules such as *do not fly your drone*

Names	Key=Value
National Park	boundary=national_park
Runways	aeroway=runway
Waterbody	natural=water waterway=*

Table 3.4: Examples of RuleMaDrone definitions based on OpenStreetMap's key value pairs.

closer than 5 kilometres to runways. An example of the zone-based adding for spatial rules can be found in Fig. 3.5.

3.2 Environmental Factors

In this section the environmental factors implemented in RuleMaDrone are discussed. We implemented a warning for three different environmental factors that can reduce flight safety. The first to be examined is the weather. The k-index and how solar activity can result in drone crashes will be explained afterwards. This section concludes with the difference between day and night being visualised on the map.

3.2.1 Weather

RuleMaDrone implemented three different elements of the weather that are important to know for the safety of a drone flight. The weather condition warns the user about rain, mist or snow on the flight location. There are certain weather conditions that are potentially dangerous for drones.

- Snow or rain could damage the electronics of most drones
- Wind can reduce the stability of drone flights and make them harder to fly
- Clouds and mist reduce the visibility of the drone for the operator

In Fig. 3.3 one can see how these different weather elements are displayed in RuleMaDrone. The icons in front of these elements provide

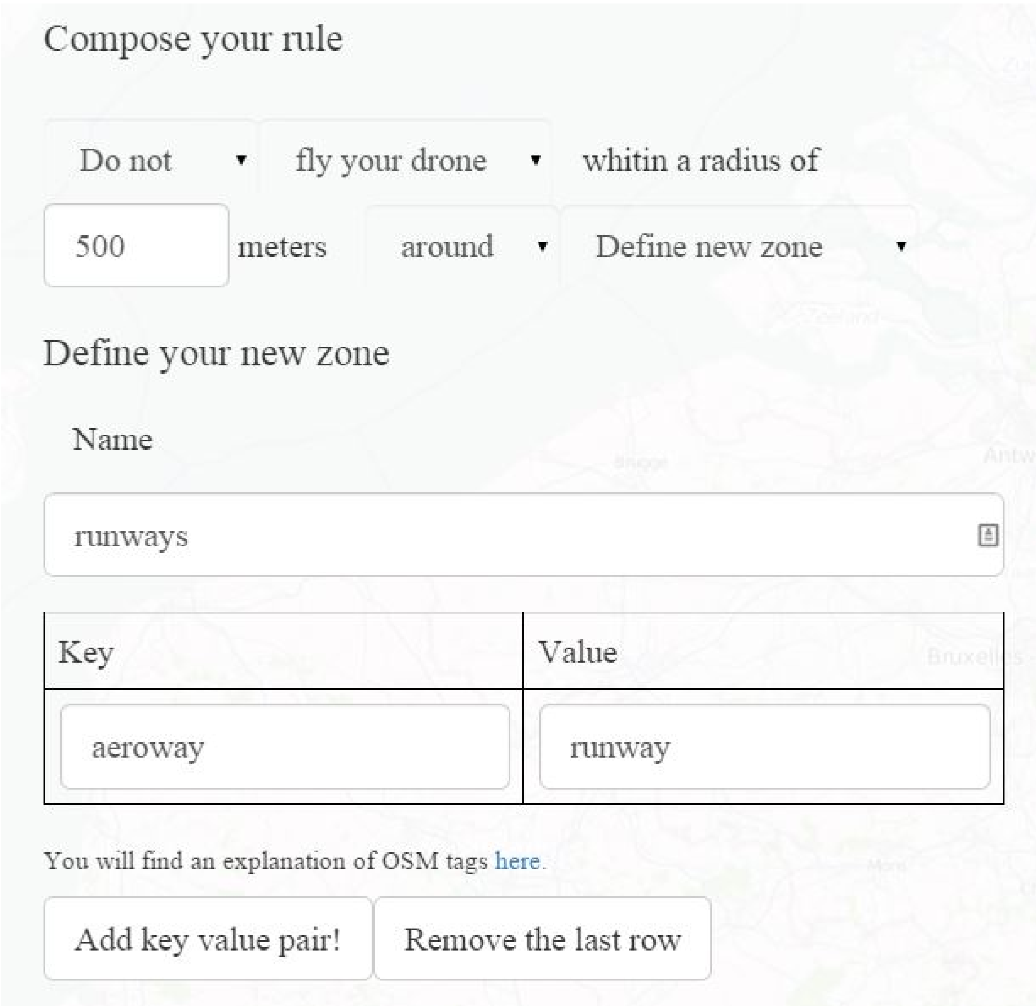


Figure 3.5: Zone-based rule adding for spatial rules

the users with instant feedback whether it is safe to fly under these conditions.

3.2.2 K-index

Modern drones have built in safety measures to avoid fly-always. A fly-away can occur when a drone loses connection with the controller. Without GPS it will start flying loose, which can result in dangerous situations and crashes. GPS is used to stabilize drone flights and prevent fly-aways. It can record the starting location, and in the case of lost connection navigate the drone back to that point.

A GPS signal can be heavily disturbed when a geomagnetic storm takes place (11). The K-index is used to characterize the magnitude of the geomagnetic storms. During a heavy geomagnetic storm the current GPS position of a drone could be totally wrong. In this case these mechanisms that should protect a drone from flying away cause the fly-away. Suppose a drone is flying somewhere in the Netherlands when there is a heavy geomagnetic storm. The drone operator loses control of the drone and it starts to fly back to its starting spot. Due to the geomagnetic storm it receives GPS coordinates from somewhere in Spain and starts flying at full speed north to its starting position in the Netherlands. Since the drone is in reality in the Netherlands, it will only fly further away from its operator. The drone will fly north until its battery is empty or it crashes into something. In this case GPS does not prevent but cause the fly-away. Because of these situations we chose to display the k-index in RuleMaDrone. Warnings concerning the k-index for the next 24 hours are displayed in the rule menu and the current k-index is also presented in the fly zone pop-up.

3.2.3 Day-Night

Drone flying at night or twilight is forbidden in some countries due to the reduced visibility. This is why a day-night distinction is made on the map (Fig. 3.6).

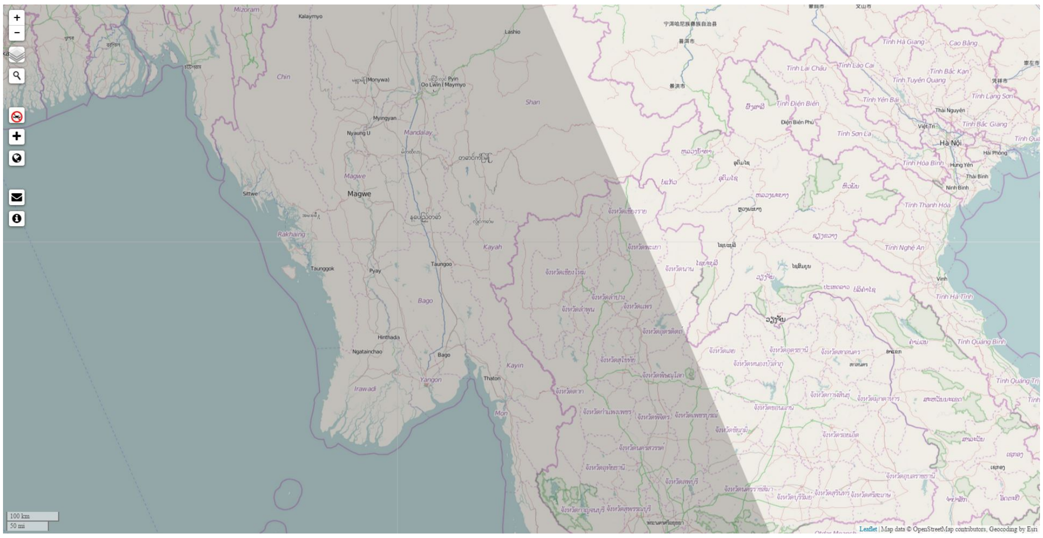


Figure 3.6: The difference between the side of the earth with daylight and without visualised in RuleMaDrone

3.3 Drone Properties

The last component of the drone safety framework are the drone properties. In the rule menu the user is provided with an option to select the drone he wants to fly. He can also choose a custom drone and fill in these properties. The properties that are currently implemented are the maximum wind speed a drone can handle, the flight range of the drone, if the drone flies on GPS or not and its weight. How these properties are used will be explained in the following section.

3.4 Fly Zone

The fly zone calculation is a feature which makes RuleMaDrone unique when compared to the tools in chapter 2. The three components of the drone safety framework in Fig. 1.5 come together in the calculation of the fly zone. Once the flying location is determinded, RuleMaDrone can be used to calculate the fly zone. The fly zone is created in such a way that no aspect of it interferes with a spatial rule on the map. It is obvious in Fig. 3.9 that the fly zone is not complete due to its intersection with a spatial rule. In the pop-up one will find information of

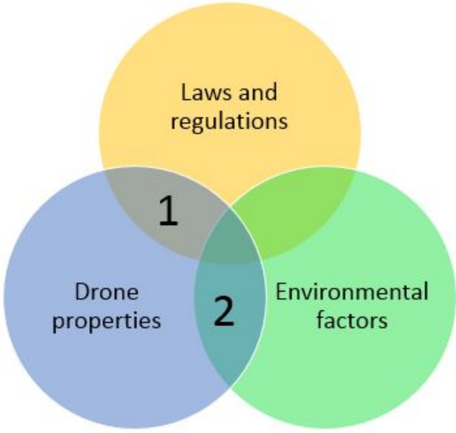


Figure 3.7: Zone 1 in the drone safety framework applies, for example, to the inference of the weight of the selected drone and the maximum weight for a drone in a country. If the weight of the selected drone is heavier, the application will issue a warning. Zone 2 in the drone safety framework applies to the inference of current wind speed and the maximum wind speed a drone can handle.

all three components. The weather information is clearly environmental information. If a warning sign is generated by one of the weather conditions depends on the drone properties, this represents zone 2 in Fig. 3.7. The list of general rules consists of additional legal information about the area where the drone operator will fly his drone but also drone specific information such as *the drone can not be heavier than ten kilograms*.

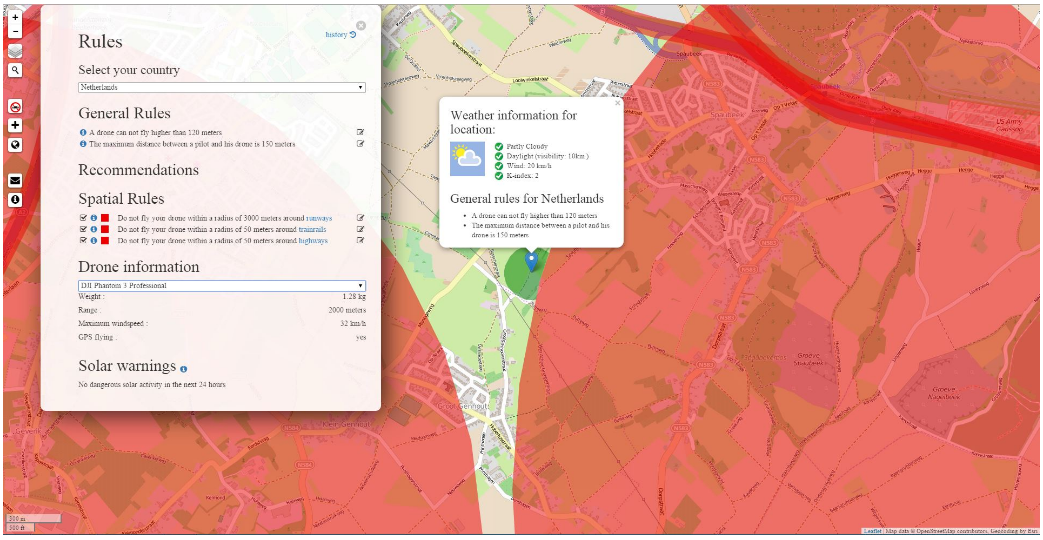


Figure 3.8: The calculated fly zone on the two-dimensional map, including the pop-up with environmental information and general rules for the country

Figure 3.9: The calculated fly zone is visualised on Microsoft Bing Maps with Cesium

Chapter 4

Implementation

The aim of this thesis was to implement a web application visualising the rules and regulations regarding professional and consumer drones. Three units were required to complete the back-end of the system: a processing unit, the database and a serving unit. These units can be seen in Fig. 4.1. Central in Fig. 4.1 is the database, rules and regulations are stored in this MongoDB database. On the left side of the database is the processing unit, which creates map layers from .lua and .osm files. On the right side of the database is the serving unit. The serving unit retrieves the rules and calculates the fly zone to send it to the applications. In this case a web application was developed, but all kinds of applications can be connected to the serving unit.

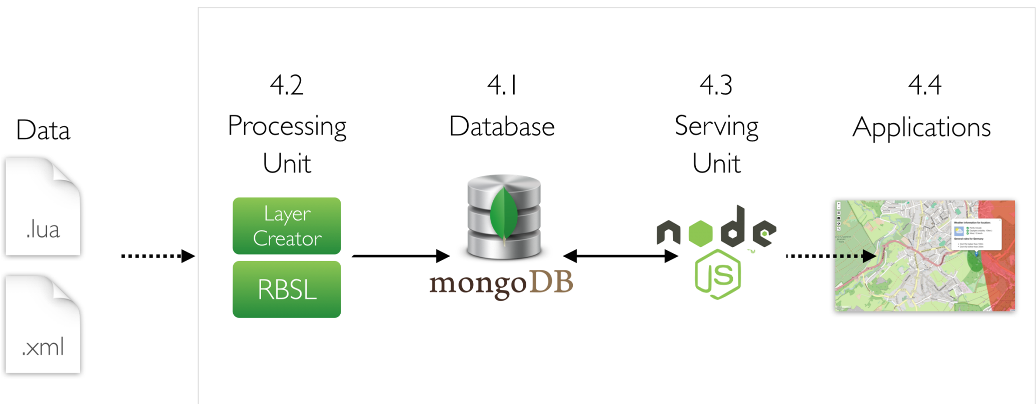


Figure 4.1: Processing Unit - Database Unit - Serving Unit

4.1 Database - MongoDB

The center of the application is a MongoDB database. MongoDB is a NOSQL document-oriented database. MongoDB has a document model. The data is stored in different collections. Each collection consists of different documents. Fig. 4.2 shows a comparison between MongoDB and a relational database. Differences between these two types of databases and the advantages of MongoDB will be discussed in the following section.

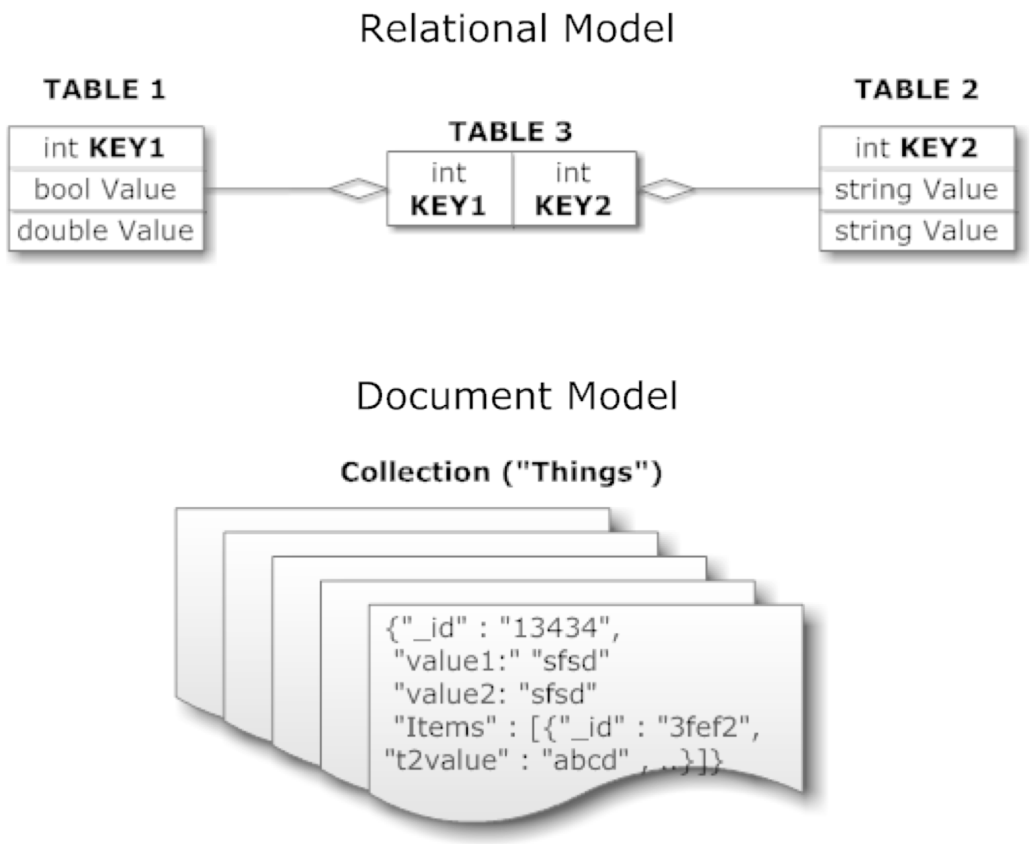


Figure 4.2: Differences between MongoDB and a relational database visualised

Rule	Max-height	Start	Stop	Distance
Only fly drones between 9.00AM and 9.00PM	null	0900	2100	null
Do not fly drones higher than 400meters	400	null	null	null
The maximum distance between a pilot and his drone is 500 meters	null	null	null	500

Table 4.1: Drone regulations stored in an SQL-table.

4.1.1 Advantages

A document-driven database such as MongoDB has certain advantages over a relational database. These advantages support our choice of MongoDB to store rules.

Agile Data Model

A MongoDB collection corresponds to a table in a relational database. In contrary to such a table, a collection does not have a strict schema for its documents. This is a very useful feature in the case of storing rules. In table 4.1 rules are stored in a relational database, this table consists a lot of null values. Every new parameter that is introduced to the rules will require a new column, which is inefficient. In a MongoDB collection these null values are omitted due to the agile data model. Not every document in a MongoDB collection must consist of the same fields. One can easily see in code 1 that only the required tags are saved. This results in a more elegant solution to store these rules.

Code 1: Drone regulations stored in a MongoDB collection.

```
[
  { "rule":"You can only fly drones between
                        9.00AM and 9.00PM",
    "start":0900,
    "stop":2100
  },
  {
    "rule":"You can not fly drones higher
                than 400 meters",
    "max-height":400
  },
  {
    "rule":"The maximum distance between
                a pilot and his drone is 500 meters",
    "max-distance":"500
  }
]
```

Compatible with JavaScript

When building a web application JavaScript is inevitable in the front-end. The choice for Node.js in the serving unit results in an application which uses JavaScript in our web application and the serving unit. When using MongoDB the results of the query's are returned in JSON objects. These objects can easily be handled by the application and Node.js in JavaScript. An abstraction layer between the database and the model is thus avoided by using MongoDB. The retrieved documents can easily be handled by the serving unit and the web application. Most other programming such as Java and C++ have a JSON parser included, so there won't be much of hassle when developing applications in other languages.

4.2 Processing Unit

The processing unit consists of a process which supports the generation of those map layers that represent spatial rules for drones. This process is completely automated. A natural language interface in the web application enables users to enter rules in natural language. Once a rule provided by a user is verified by an administrator, the layer is generated by the processing unit. In this section the work flow in Fig. 4.3 and the drawbacks of this process will be discussed.

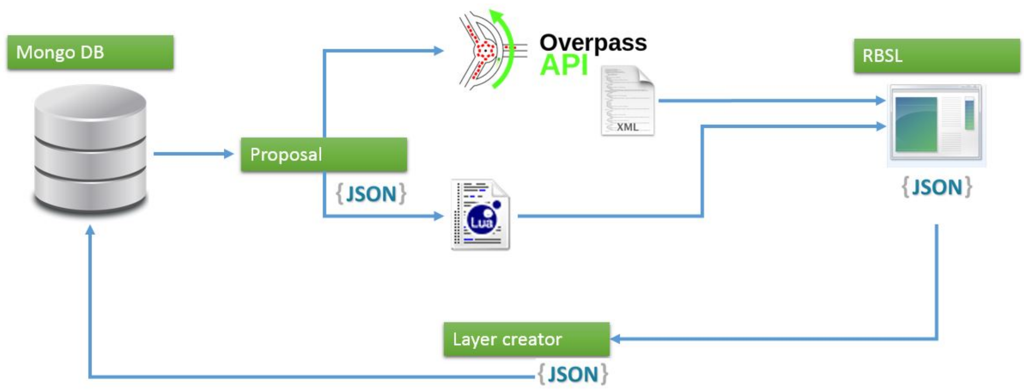


Figure 4.3: Visualisation of the process that converts a natural language proposal to a rule layer

4.2.1 From Rule Proposal to Tagged OSM features

Proposed rules are stored in the database as documents. These documents include all the information required to create a layer that visualises a spatial rule. The information such documents contain is shown in table 4.2.

Starting from such a JSON object it is required to create a lua file. These lua files can be used to extract the correct OpenStreetMap features and tag them with a certain rule, this is done with RBSL. These features are then returned in a GeoJSON file, which is used in the layer creation. However, it is possible to use RBSL with OSM files of any file size to extract the features. It was chosen in this version to use smaller OSM files downloaded from the Overpass API. Overpass has a

Field	Value
Rule	Is it recommend or forbidden to fly a drone in this area
Why	The rule in natural language
Zone	Definition of the zone where this rule is defined existing out of OpenStreetMap tags
Link	An URL of the website where the rule can be verified
Radius	An optional value specifying in which radius around the zone the rule applies to
Ruletag	This tag can be for example drone_flying or alcohol_consumption. This makes sure the application can be extended with other space usage rules.

Table 4.2: The information stored in user proposal for a rule.

query system to retrieve only the required features of a certain country. This resulted in a better execution time for the RBSL application and a better execution time in general.

4.2.2 Layer Creation

RBSL does extract the correct features from OpenStreetMap but does not create a layer of these features. With RBSL a JSON file consisting GeoJSON features is retrieved, this file can be converted into a map layer. To do this two steps are executed:

1. Buffering
2. Unioning

4.2.2.1 Buffering

Before combining all these features into one layer, computing the buffer of each of these features is required. We buffer the retrieved features for two reasons.

Applying radius tag It is possible for a user to add a radius component to a rule e.g. *Don't fly within a radius of 1500 meter of runways.* Then all features extracted with RBSL are tagged with a radius of 1500 meters. These buffers are, however, not yet created, thus we buffer the features here to create them.

Remap on polygons An RBSL file is able to return features that are points, lines or polygons but to work correctly with RuleMaDrone all the features are required to be converted into polygons. This is why the lines and the points are converted into polygons by buffering them by one meter, even if there is no radius tag. The idea that there are only polygons in each layer simplifies the calculation of the fly zone and makes the use of a draggable marker in the front end possible. Dropping a marker exact on a point or a line is difficult.

4.2.2.2 Unioning

An important component of the layer creation is the union of the buffers into one layer. In this thesis we searched for the fastest method to union all these buffers together into one layer. Three strategies were implemented during the development of RuleMaDrone: in this section the sequential, pairwise and cascaded union techniques will be discussed. The input of all these algorithms is the list of buffers computed in the buffering step, see 4.2.2.1.

Sequential unioning The first method that was used during implementation is called the sequential union. The first element of the buffer list is used as the union container. Then sequentially all the other buffers are unioned with this container. In Fig. 4.4 the union of buffers 1,2,3 and 4 in the top left are calculated. Polygon 1 is chosen as the union container. Then, we sequentially union polygon 2,3,4 onto this container.

This approach has the advantage of an easy implementation. The drawback of this technique is that with every union the container becomes larger. Consequently it gets computationally harder with every step to calculate the union. The orange line on Fig. 4.6 makes this trend visible.

Algorithm 1: Sequential union

```
procedure Sequential union
    unionContainer ← buffers[0]
    buffers.remove(0)
    while buffers not empty do
        unionContainer ← union(buffers[0], unionContainer).
        buffers.remove(0)
    layer ← unionContainer
    return layer
```

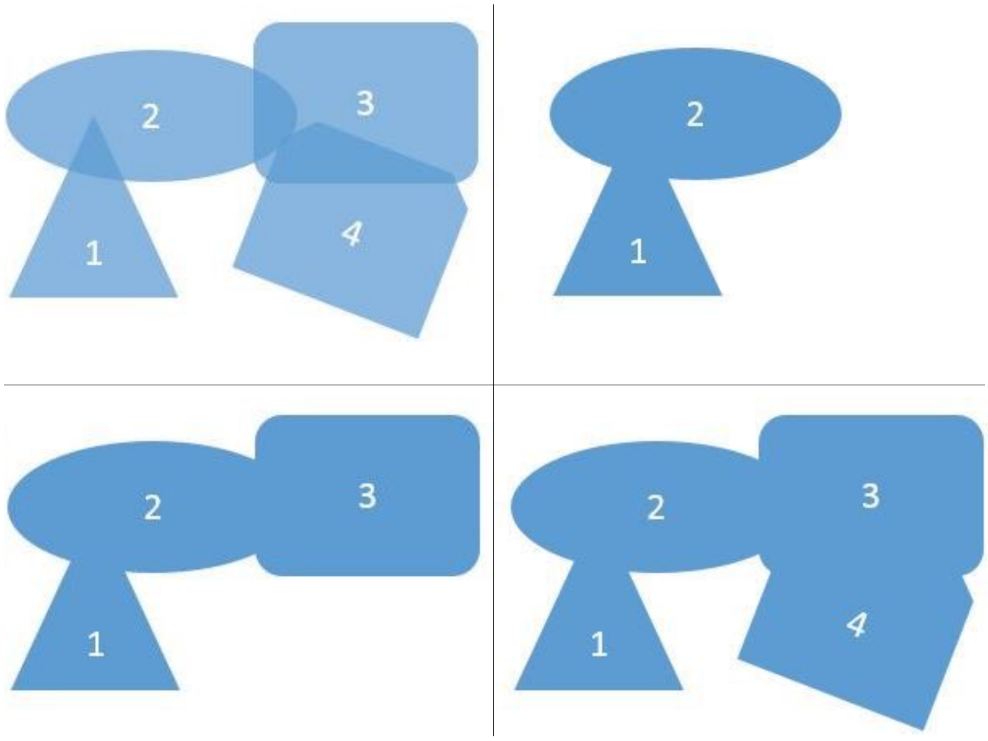


Figure 4.4: Visualisation of the sequential union

Pairwise unioning In order to solve the problem of the previous approach, a pairwise union was applied to the list of buffers. This approach loops until there is only one element left in the buffer list, this element becomes the rule layer. In every iteration the union of the first two buffers is calculated and added to the back of the list. In Fig. 4.5 the union of the buffers 1,2,3 and 4 needs to be calculated. When done pairwise the union of buffer 1 and buffer 2 is calculated first. Next, the union of buffer 3 and buffer 4 is calculated. Finally the union of buffer12 and buffer34 is calculated. By looking at Fig. 4.6 one can see that the time it takes for each union in the pairwise algorithm is remarkably shorter for almost all unions compared to the sequential union. In the sequential union, the union container grows rapidly in size. In the pairwise approach we do not have one big union container, the features that must be unioned every time are smaller for longer period because they grow in size more slowly. A disadvantage of this approach is the fact that the last union operations can take a very long time because these are two very complex polygons instead of one in the sequential approach. In general the pairwise approach is a lot faster than the sequential approach as can be seen in table 4.3.

Algorithm 2: Pairwise union

procedure Pairwise union
  while length of buffers is larger than 1 **do**
    $temp \leftarrow$ **union**$(buffers[0], buffer[1])$.
    $buffers.remove(0)$
    $buffers.remove(1)$
    $buffers.push\_back(temp)$
  $layer \leftarrow buffers[0]$
  return layer

Cascading union[1] The shortcomings of the previous approaches were due to the fact that very large geometries were required to be unioned in the end. This method omits the calculation of large unions by dividing the buffers into smaller parts. The first step of this approach is to calculate the minimal bounding box of all buffers combined. This

[1] `http://lin-ear-th-inking.blogspot.be/2007/11/fast-polygon-merging-in-jts-using.html`

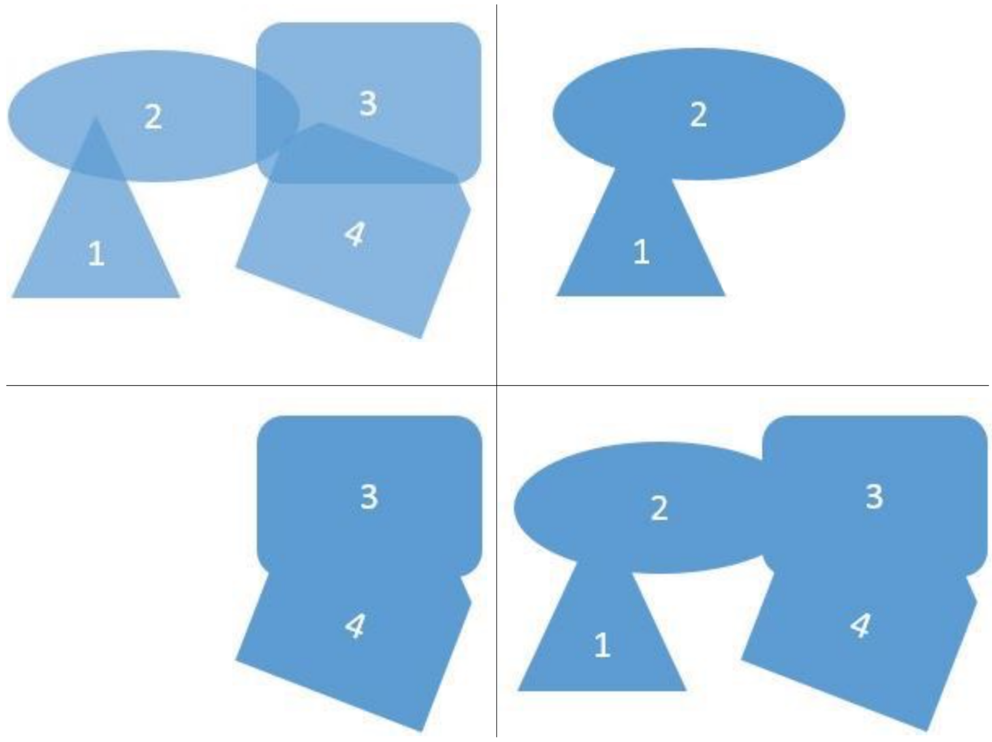


Figure 4.5: Visualisation of the pairwise union

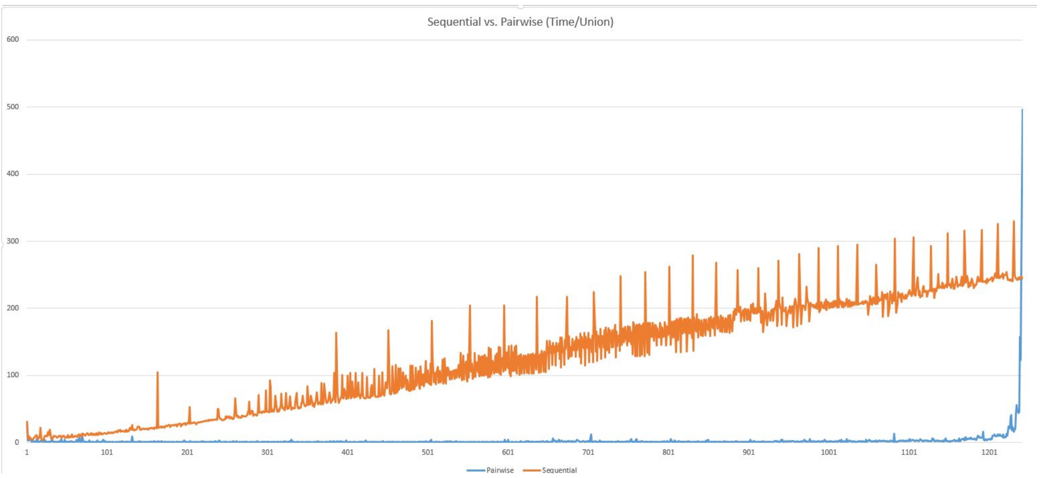


Figure 4.6: Comparing the time per union for the sequential approach (orange line) with the pairwise approach (blue line). To do this we used the rule layer of the rule : *do not fly in a radius of five kilometres around runways in France*

bounding box is recursively broken down into four polygons which are equal in size, we will call those segments. Every segment holds a list of polygons with which it overlaps. This process is repeated until there are 65536 segments left with a small list of buffers overlapping the segment. However, since segments with an empty buffer list are discarded, there will be most likely far less than 65536 segments left.

Now we iterate two times over the list of segments. In the first iteration the intersection between the segment and each of the buffers overlapping it is calculated. The list of these results is now saved in the segments instead of the list of overlapping buffers. Now in every segment there are only partial buffers left that do not overlap with partial buffers of other segments, just as in Fig. 4.7 number 2.

In the second iteration through the list of segments all the partial buffers of each segment are unioned. The results are combined into a multipolygon. Although it would be possible to union these resulting polygon back together, it was decided not to because of efficiency reasons when calculating the fly zone.

The major advantage of this approach is that it is much faster than the two previous approaches because of the smaller polygons required to be unioned. Another advantage is that the division into segments have shown to be very useful when calculating the fly zone (4.3.1). In Fig. 4.7 one can see how the cascaded union works in a simplified form. On top row the original buffers are divided in sixteen segments. The segment with a red border consists of three partial buffers. On the second row one can see how the partial buffers in each segment are unioned together. The last image shows the final result. Just as in the application the pieces are not one big unioned polygon but a multipolygon of different partial buffers. Because these buffers are lying next to each other, it is not visible in the application.

Rasterizing Another method considered during implementation was to convert the buffers expressed with vectors into a raster representation. This would result in a much faster way of buffering and unioning the buffers. The reason this approach was not implemented is because of resolution loss. When converting the vector representation into a

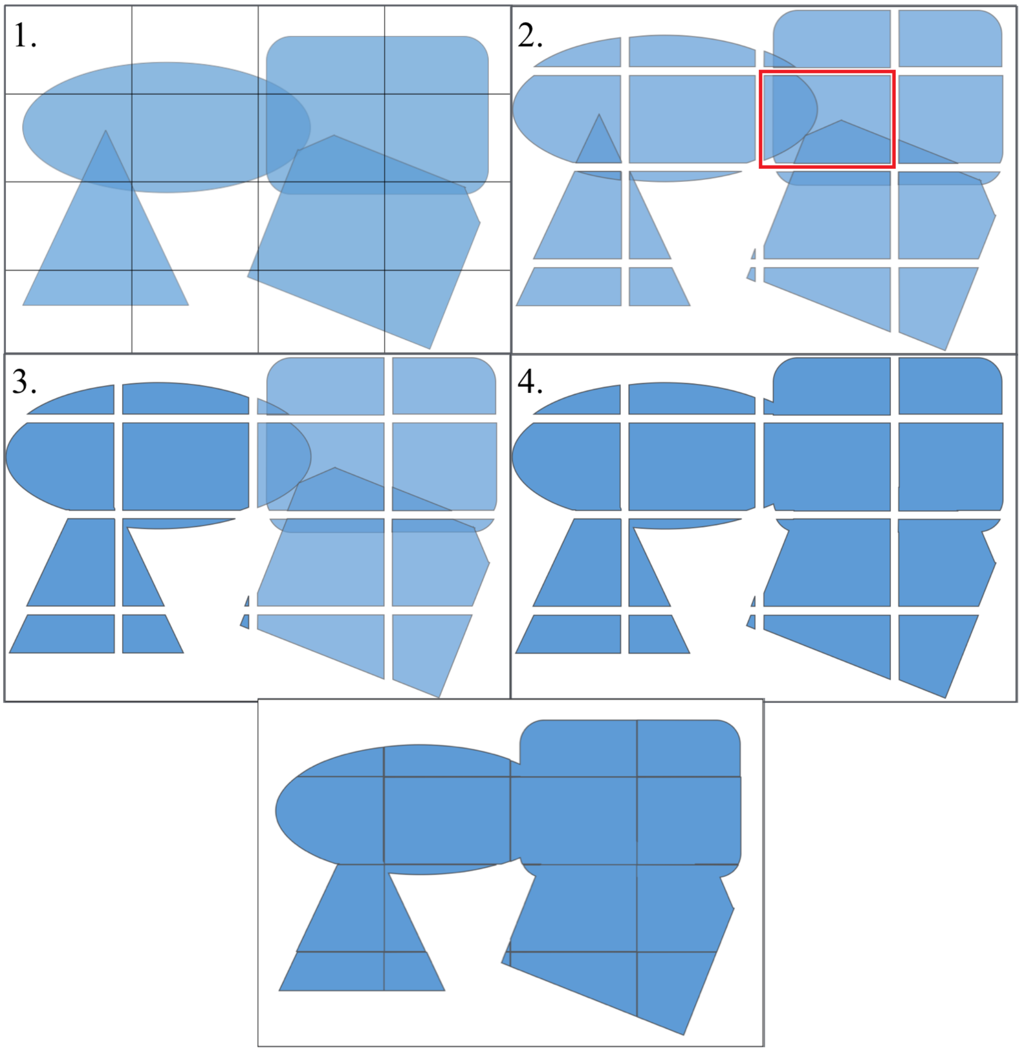


Figure 4.7: Visualisation of the cascaded union

raster representation some details will get lost. Since this is an application visualising drone laws we cannot afford the application to lose precision.

Benchmarking The benchmarks were executed on an Intel Core i7-3630QM CPU. Three different rules were selected:

- *Don't fly in a radius of 50 meter's around Dutch roads that have a speed limit over 60 km/h.*
  40 922 buffers with a lot of overlap.
- *Don't fly in a radius of 8000 meters around Canadian runways.*
  650 buffers with little overlap.
- *Don't fly over Belgian military zones.*
  134 polygons with no overlap.

	Dutch roads	Canadian runways	Belgian military
Sequential	>32h	38 sec	4 sec
Pairwise	29h	1.624 sec	0.715 sec
Cascaded	3 min	1.536 sec	0.724 sec

Table 4.3: Results of benchmarking.

The cascaded union is much faster when there is a lot of overlap, otherwise the difference between the cascaded union and the pairwise union is negligible. The sequential approach is in all cases the slowest.

4.2.3 Drawbacks

Large files However, RBSL does handle very large files, it is not yet possible to add a rule such as *Do not fly in 30 meters of buildings* which applies in Canada. The reason for this is that overpass-api does not handle very big queries like this. When posing such a query to the server, it results in a time-out caused by the big amounts of data it has to handle.

OSM relations OSM relations are not yet a part of the output because RBSL does not handle relations yet.

4.3 Serving Unit - Node.js

The server-side of the application is implemented in Node.js combined with express. The serving unit is a restful API which is called by the front-end. The reasons for this implementation choices are:

Integration It is possible for other developers to integrate the rules collected by RuleMaDrone in their own application. A drone manufacturer, for example, is able to make use of this service to visualise space usage rules regarding drones in the smartphone controller application.

Event loop The main task of the serving unit is to retrieve layers from the database and serve them to the client. The Node.js event loop handles these situations gracefully. Node.js does not create a thread for each request but works with just one thread. When it receives a request to retrieve something from the database, it sends the query to the database and goes on with the loop. Once the database is finished the callback is executed. The event-loop thus never *waits* for a heavy read or write operation to finish. It works the same with API calls, which we also use frequently in this application for retrieving weather data or reverse geocoding information.

The event loop described as a major advantage poses also the greatest challenge in working with Node.js. Because when this loop slows down, all users experience the slow-down. During the work on RuleMaDrone two major causes of slowing down this loop were identified.

Serving static files When this loop must handle the usual web requests such as sending html, css, javascript and other file formats the loop clogs because a lot of files must be sent each time a client connects to the homepage. To solve this issue we used Nginx. Nginx was used as a reverse web proxy in between the client and the Node.js server, see Fig. 4.8. Every time a request for a static file arrives, Nginx handles the request, otherwise the request is handled by Node.js.

Calculating fly zone The calculation of the fly zone was a computational heavy process. It took up to several seconds to calculate this fly zone, which slowed down the server enormously. Some optimizations on these calculations were found to minimize this impact. These optimizations are explained in 4.3.1.

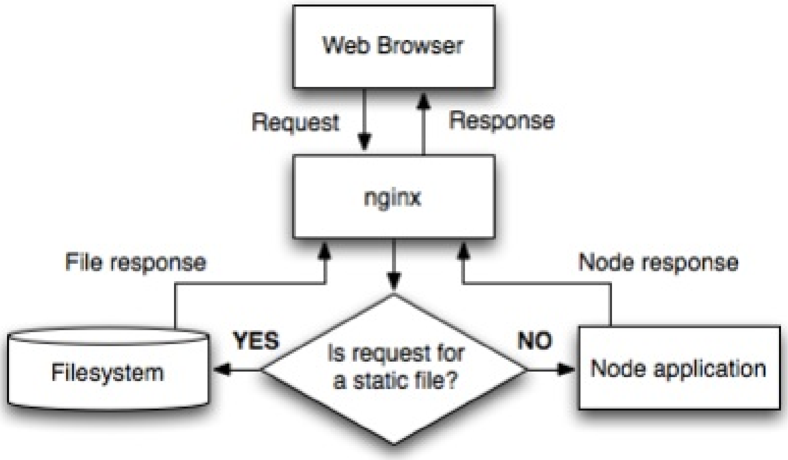


Figure 4.8: Ngix as a reverse proxy

4.3.1 Flyzone Calculation

The calculated fly zone depends on the three pillars of drone safety which are the environmental factors, drone properties and laws and regulations. The factors included in these pillars combined with the location of the drone operator determine the size and form of the fly zone.

Retrieving weather data The API of *World Weather Online* is used to retrieve the weather data.

Retrieving general rules for the country First Google's reverse geocoding API is used to determine the country in which the marker is dropped. After that the general rules for the country are retrieved from the database. These can be used to determine the radius of the fly zone and do some inference with the drone properties and the weather to determine if a safe flight is possible.

Calculate the area This process consists of a few steps:

1. The radius around the point where the user placed the marker is calculated. The radius can depend on the visibility due to the weather (environmental factor), the regulation in the country (laws and regulations) and the range of the drone (drone properties). The buffer with the determined radius is called the initial fly zone.

2. If there are certain spatial restrictions within this radius, these have to be cut out of the initial fly zone. The naive approach to do this is $flyzone-(layer1-layer2-...-layern)$. This is the slowest part in the calculation, so it was required to perform some optimizations to prevent it from clogging the Node.js event loop.

3. Check if due to the removal of spatial restrictions out of the fly zone some parts became unreachable and remove these parts. When the fly zone is divided in different polygons, we check in which polygon the marker was placed and only keep this polygon. This can easily be done with the point in polygon operation. If the marker does not fall within one of the fly zone polygons, the marker was place in a spatial rule where drone flying is prohibited and there is no fly zone.

Optimizations

Since the calculation of the fly zone is the bottle neck of the application, there was a need for optimizations. Several of these were tested during the development of RuleMaDrone.

Point in polygon in MongoDB An optimization that failed its purpose was to determine if the marker was placed in a red zone by doing a point in polygon on the layers before calculating the intersection in MongoDB. We supposed it would be faster to do the simple point in polygon operation than all these intersections. But when we tried this the results were disappointing. Because of the size of the layers the operation was not very fast and this caused a major slow-down when the marker was not placed in any of the red zones. When it was placed in the red zone, the gain was non-existent.

Using intersection before erase Using the library Turf we discovered that the operation *erase* was rather slow to use on a big layer. It was faster to calculate the intersection of the initial fly zone with the big layer first and then remove this intersection from the fly zone. When comparing both methods for calculating a fly zone in France with one layer turned on: *Do not fly in a radius of 5 km of runways* we found out that just erasing the layer from the fly zone took 418 ms while first intersecting the layer and the fly zone and then removing the fly zone took 361 ms. This can make a major difference when a lot of requests are received and a lot of layers are turned on.

Reduce vertices before intersecting The initial fly zone has a lot of vertices because it is a circle. A circle can easily have up to 36 vertices. When intersecting this with a layer with a few hundred vertices this becomes computationally heavy. This optimization reduces the amount of vertices by doing the intersection on the bounding box of the initial fly zone which has only four vertices. Then the result is removed from the initial fly zone in order to get the final fly zone.

Working with partial layer polygons The last optimization was due to the optimization in layer creation. In layer creation we divided the layer in $4^7 = 65536$ different segments. Each of them holds a different part of the layer. Each of this parts still has its segment still in its properties. The minimum bounding box of the whole layer is also stored with the layer. Now a few simple checks can be done to prevent heavy calculations. If the bounding box of the initial fly zone does not overlap the bounding box of the complete layer, we do not have to calculate any intersection for the whole layer. If the bounding boxes do overlap, every segment can be checked against the bounding box of the initial fly zone for overlap. Most segments will not overlap and so we reduce the amount of intersections. The calculation of the fly zone with the layer *Do not fly in a radius of 5 km of runways* in France reduced from 361ms to 20ms because a lot of intersections are omitted.

4.3.1.1 General Algorithm for Fly Zone Calculation

In Fig. 4.9 and Fig. 4.10 the general algorithm of the fly zone calculation is shown. Fig. 4.9 displays an image of a chosen point for the user to fly and a layer of forbidden zones. In this example the rule could be *don't fly within 1 kilometer of a runway*. We suppose that there is a rule determining the maximum distance between the drone and the operator. The circle is drawn but overlaps with some red parts. Before we start removing this parts of the circle we draw a bounding box of the circle and select all the parts of the layer that intersect with this bounding box. We calculate the circle minus the parts that overlap with the bounding box and the result is the fly zone. In Fig. 4.10 the same process is done except for the image where the point in polygon operations are performed. These operations are done to ensure that maximal one polygon is left, the one which contains the point. The other polygons are removed. This happens with the upper left part of the fly zone.

4.4 Application

The application created within this thesis is a web application. The front-end communicates with the serving unit discussed in 4.3. In this section we will examine the major components providing value for our application. To implement the application we used Leaflet for the two-dimensional map, Cesium for the three-dimensional map and Jade for templating the HTML files. The application is kept 'dumb', no calculations are made in the application itself. This is because if an application is built on another platform, the developer can, for example, simply use the server to calculate the fly zone.

4.4.1 Leaflet

Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps. It is developed by Vladimir Agafonkin with a team of dedicated contributors. Weighing just about 33 KB of JS, it has all the features most developers ever need for online maps. [2]

[2] `http://leafletjs.com/`

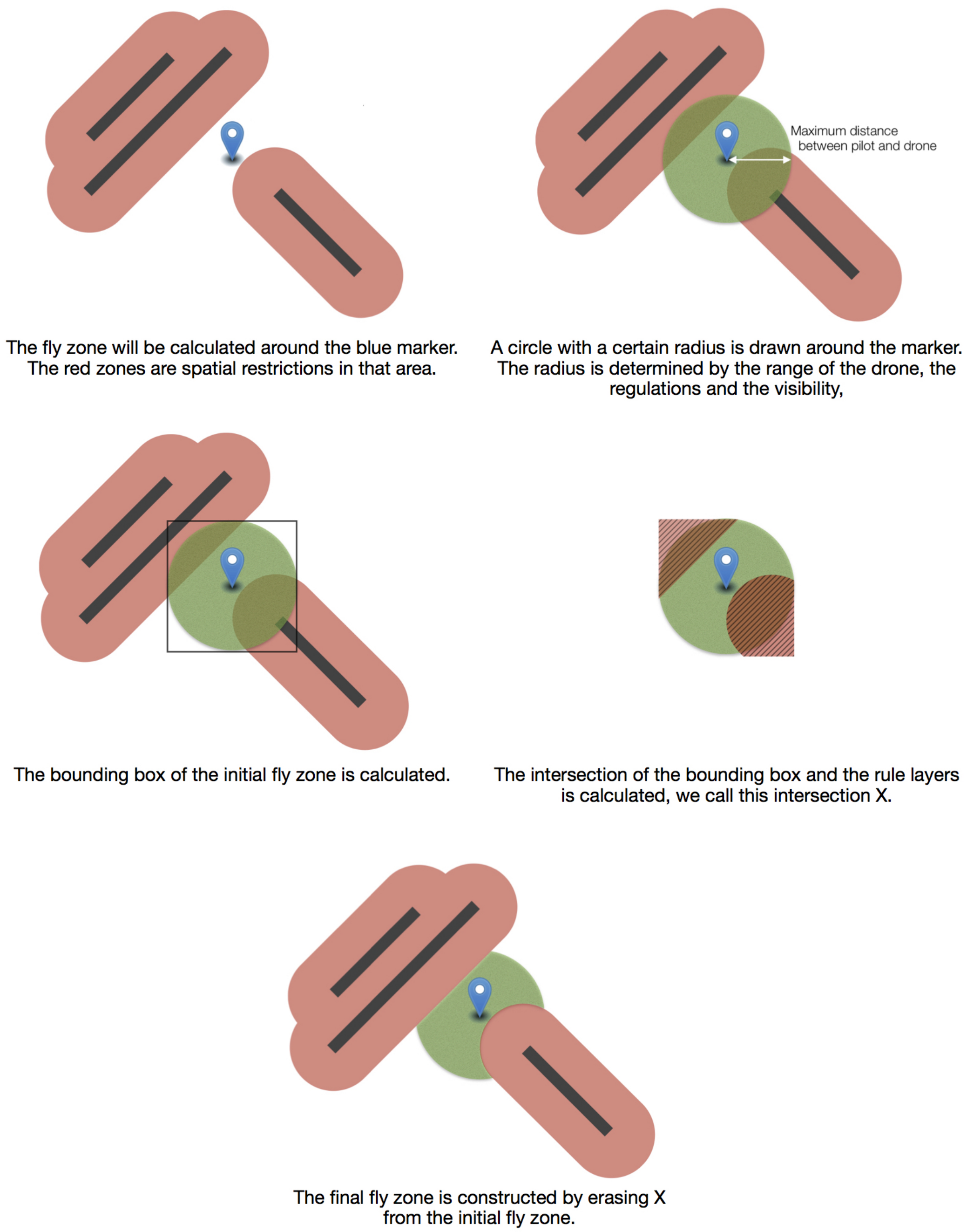


Figure 4.9: Scenario 1 for calculating the fly zone

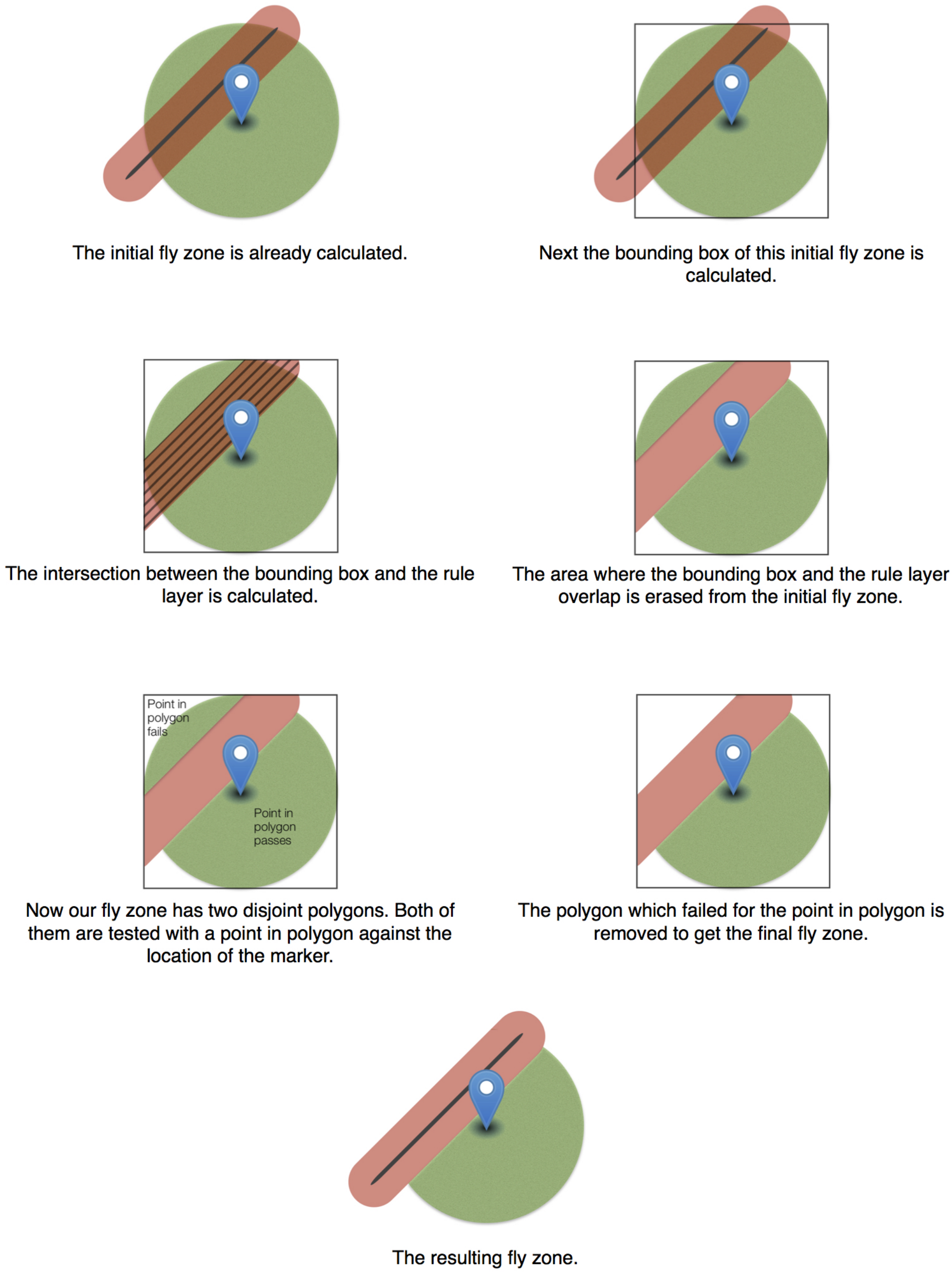


Figure 4.10: Scenario 2 for calculating the fly zone

Leaflet consists of all the components that are required to visualise our map with the spatial rules on it. First of all it was possible to use the tilelayer of OpenStreetMap, this prevents the mismatch between OpenStreetMap data and Google maps. In addition to this Leaflet aligns beautiful with our choice of GeoJSON in the server. Leaflet provides an easy manner to visualise GeoJSON.

4.4.2 Cesium

At first the Google Earth Plug-in was used for the three-dimensional view. This, however, caused several difficulties for the implementation. One of it was a known bug that a plugin crashes once the div containing it was hidden. [3] Another inconsistency with all other components is that Google earth used KML-files to visualise the data, so it was required to write a converter from GeoJSON to KML-files. The last problem with the Google Earth Plug-in was that the top of the fly zone did not follow the terrain beneath it, which is a known issue.[4] With the deprecation of the Google Earth Plug-in and plug-ins in general we had to make a switch there.

Cesium was a lot easier to integrate into our application for several reasons. First of all Cesium is not a plug-in, which makes it still usable on all major browsers. Secondly Cesium is able to use GeoJSON for displaying the layers. It had two other problems: we did not find how to turn on the 3D terrain in Cesium and the controls of Cesium are not intuitive.

4.4.3 Jade Templates

Jade templates were used because we decided that it had to be possible for the application to be turned into a visualisation tool for other space usage rules such as *no dogs*, *no smoking* or *no fishing*. With Jade templates we could easily remove the parts which have no use for most other space usage rules such as the solar activity and the possibility to

[3] `https://code.google.com/p/earth-api-samples/issues/detail?id=141`
[4] `https://code.google.com/p/kml-samples/issues/detail?id=432`

choose a drone. With Jade templating the HTML page is rendered on the server and send to the application.

Chapter 5

Evaluation

During the development of RuleMaDrone, we conducted two studies. The first study consisted of an online survey to collect quantitative and qualitative data on the need of such an application. The second study was exclusively conducted with participants who deal with drones in their daily professional life. The goal was twofold: collect qualitative data on the usability and quantitative data on how they would like to see this application evolve in the future. After the release of RuleMaDrone we used the data of Google Analytics for a third study.

5.1 Survey

The goal of our first study was to research the demand for this application. We created an online survey which would reveal whether the application was perceived as useful. The users were asked for their opinion on drones in general, the application as a whole and each of the features individually.

5.1.1 Participants

After being online for nearly two months 203 participants completed the entire survey. The majority of participants were students under the age of twenty-five. Seventy-three percent of the participants were male. The main nationality for people that filled in this survey was Belgian with 194 participants, there were also four Dutch and five with other nationalities.

5.1.2 Results

In this section the results of the survey are discussed. We first examine the study participants' opinions, their concerns about drones and their knowledge about the drone laws in their country. Since our tool can be regarded as a planning tool we then discuss the flight planning of the drone operators. Lastly we look into the answers participants gave about the usefulness of RuleMaDrone and its features.

5.1.2.1 Opinion on the Rise of Domestic Drones

Most survey participants were moderately positive on the rise of domestic drones. On a scale from 'one' (very negative) to 'five' (very positive) over seventy percent chose 'three' or 'four'. The most negative option, 'one', was chosen least.

5.1.2.2 Concerns

Two different questions were asked to understand the user's concerns regarding drones. The first question was an open one where they could express their concerns. We saw that seventy percent of the participants were indeed worried about some topics.

The law was listed as a concern by six of the fourteen drone owners that completed the survey. These law-related comments can be found in table 5.1.

The second question gave the users four options as their major concern: nuisance, privacy, safety or none at all. When given these options only four percent had no concerns at all. This showed a tremendous difference in concern between the drone owners and the other participants. Seventy-eight percent of the drone owners chose safety as the main concern. While only fourteen percent was more concerned regarding the privacy issues (Fig. 5.1). The participants without a drone were far more concerned about their privacy: a majority of sixty percent chose privacy as main concern. Safety was second with only thirty percent. Nuisance was only chosen by five percent (Fig. 5.2).

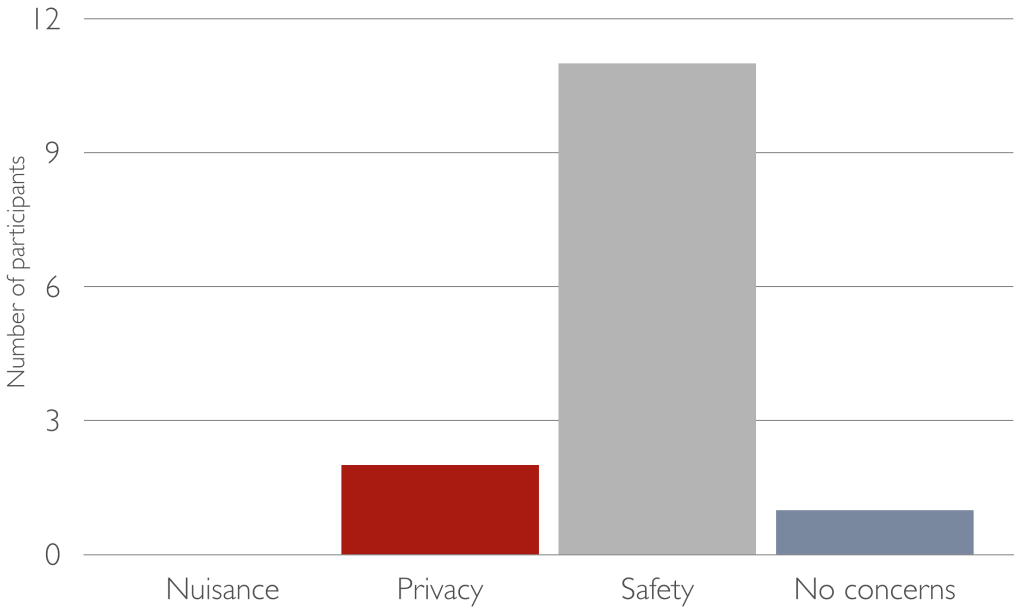


Figure 5.1: Concerns of the drone owners

It is not allowed .
Mostly I don't know what the height limits and such are .
Of course. We don't have a law in Belgium.
Legislation that becomes operative in 2015 is too limited for professional service providers. More flexibility in the airspace is needed for the professional UAS companies.
There is no legislation or inspection about drone flying. Flying a drone, no matter how small, can be very dangerous and must be monitored and regularised.
Yes, everybody needs a license to fly them .

Table 5.1: Law-related concerns mentioned by drone owners

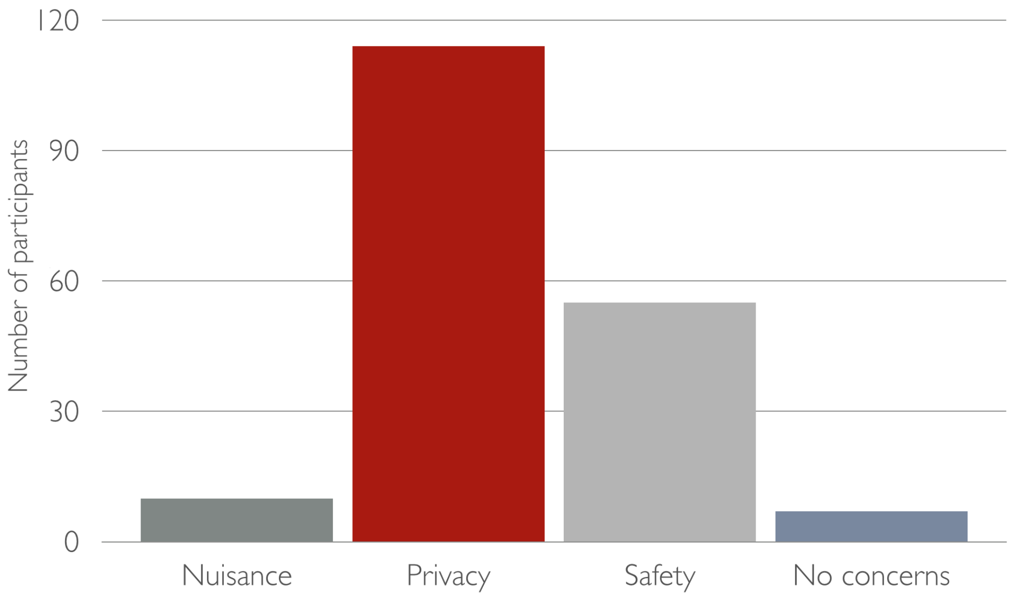


Figure 5.2: Concerns of the none drone owners

5.1.2.3 Knowledge of Drone Laws

The overall rule knowledge regarding drones is insufficient. Almost seventy percent of the participants acknowledged that they have no knowledge about drone laws in their countries, as can be seen in Fig. 5.3.

5.1.2.4 Flight Planning

Since the next features of our application depended on the results of this survey and the interviews with professional drone operators, the drone owners were asked about their preparations before a flight. Two out of the fourteen participants with a drone do not perform any checks before flying their drone. Another two mentioned they have a check list to complete before take-off, but did not mention anything on the check list. By filtering these four responses we got ten useful ones from drone owners. Six of them declared checking the weather, especially wind speed. Five mentioned looking on a map and checking the location of the flight. Material checks were also named a lot, especially checking

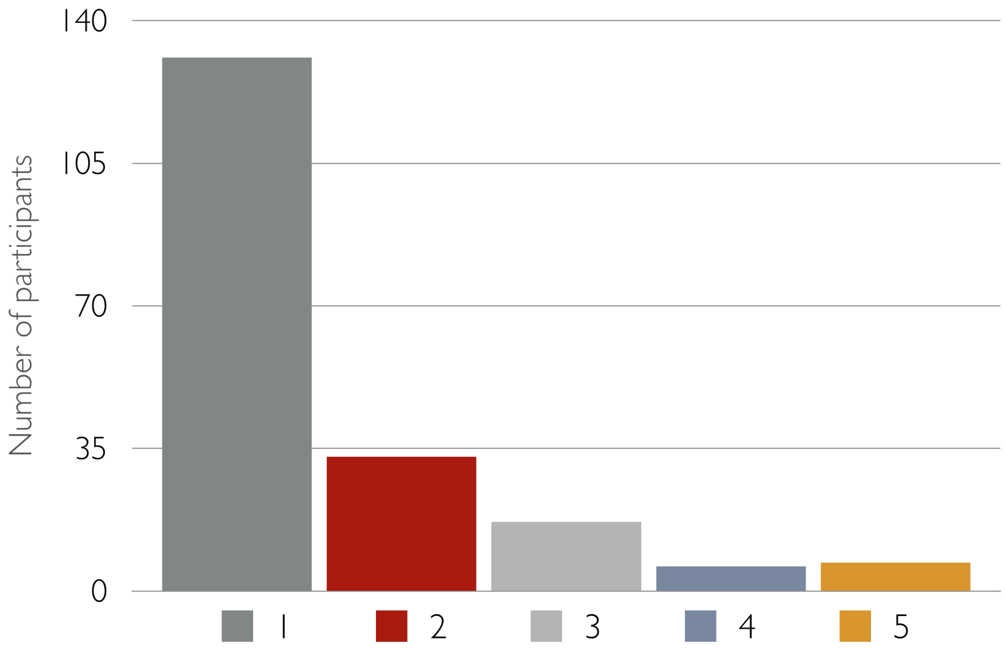


Figure 5.3: Rule knowledge of civilians about drones

the battery. Also checking the K-index was mentioned, which was later included in RuleMaDrone.

5.1.2.5 Feature Usefulness

To find out the usefulness of the application we asked the participants to rate how useful they found every feature included in the application. The rating scale went from 'one' (useless) to 'five' (major importance). The results are shown in Fig. 5.4 and Fig. 5.5. The more traditional features such as location search, fly zone calculation and weather info scored high in this study. These features are all rated with more than 75 percent of fours and fives.

5.1.2.6 Would You Use RuleMaDrone or a Comparable Tool if You Had a Drone?

Eighty-six percent of the participants answered this question with yes. We then studied the fourteen percent of the users who would not use

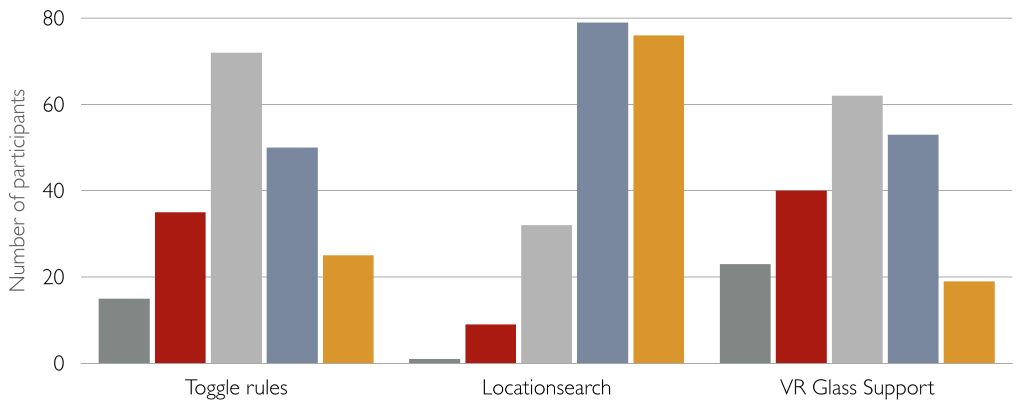


Figure 5.4: Usefulness of the features rated on a scale from 1 (useless) to 5 (major importance)

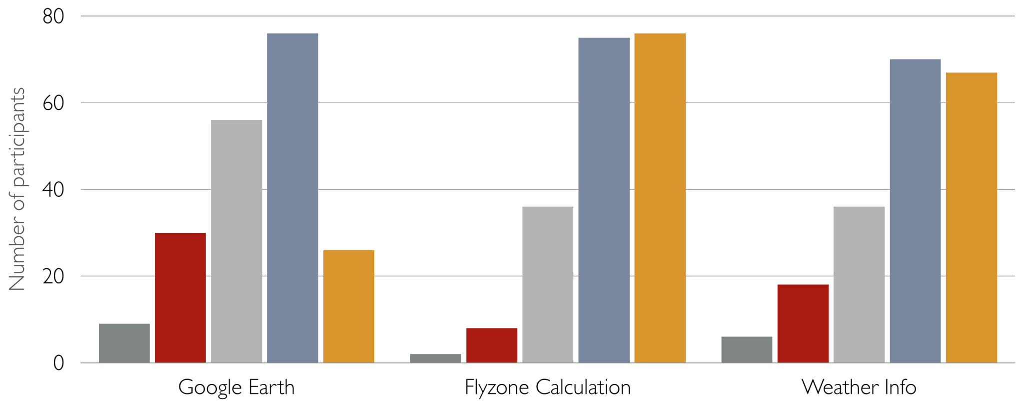


Figure 5.5: Usefulness of the features rated on a scale from 1 (useless) to 5 (major importance)

such a tool. Remarkably the knowledge of the law in this group is not good at all. None of these participants rated their law knowledge with ’four’ or ’five’. Eighty percent actually rated their own knowledge about drone laws with ’one’. One person who would not use the tool actually owns a drone, but said since he only flew at home or at the beach he had no use for this. Ten of the twenty-seven people gave as reason that they did not own a drone, which was actually not required for a ’yes’ to this question. Some other responses were: *I have no use for it, I don’t need it* and *not useful*. There was also one person who thought that in general people would neglect the rules.

5.1.2.7 Extra Features

The last question of our survey was which features the users would like to see included. Most of them were out of scope and regarded as eventual future work but we still present a few of the features proposed.

- GPS for drones
- Show drone live on map, warnings when flying into forbidden zones
- Limit fly zone with reducing battery power
- Relief map
- ATC zones
- Frequency warnings
- High obstacles
- Integration into the controller
- Social tools for drone flyers (geocaching, commenting on locations and recommending locations)

5.2 Interviews

The second study we conducted was an on-site interview with professional drone pilots. They were asked about how they prepare flying their drone and which rules they thought they had to obey in Belgium. Thereafter they had the opportunity to use and rate the usability of the application.

5.2.1 Drones@PXL

On December 19th, 2014, the team leader explained the current situation regarding drones in Belgium. According to him his research team is restricted because of the Belgian rules. It is only legal to fly if you have permission of the DVLG. Unless you are flying in an hobbyists club or you are less than 14 years old. That is why he always flies inside buildings where no other people are endangered. For flying with a camera a permission is always needed. Important tools to see if you can fly legally are low air charts. A drone flyer must also take the regulations regarding security camera's (kb 02.07.2008) and protection of the personal privacy (kb 13.02.2001) into account. He also said to look at the frequencies for the controller and the video stream.

Furthermore he summed up some of the most important organisations regarding the drone regulation:

ICAO - International Civil Aviation Organisation

EASA - European Aviation Safety Agency

BEUAS - represents the interests of all the Belgian enterprises and institutions, which are active in the unmanned aviation sector.

DGLV - regulates Flemish aviation

Belgocontrol - An autonomous public company in charge of the safety of air traffic in the civil airspace for which the Belgian State is responsible.

VML - Vlaamse Modelbouw Luchtvaart

5.2.2 Koptermann

On January 23, 2015 we spoke to the founders of Koptermann. To fly in Belgium companies have to ask permission at the DVLG. But flying in

CTR's (around an airport) and LFA's (zones where military aircrafts can fly low) is not possible. These rules already cover half of Belgium. Even outside these black zones, they still don't get a permission for flying in cities, in the evening, and so on. It would be useful to include these zones next to NOTAMS (Notice to Airmen) which are temporarily flight restrictions in some regions due to high building constructions and so forth.

With regard to RuleMaDrone we were alerted to the development of a similar application by another company. At Koptermann we were told more about that application, in particular that it is created with the needs of commercial drones in mind, but too complicated for usage as a hobby drone pilot. They recommended to keep working on consumer drones and to simplify the tool so that the users can benefit from it even before they buy drones.

A usage scenario the Koptermann founders gave is the following: If a hobbyist has decided to buy a drone and has a few options left, he visits RuleMaDrone and inserts the serial numbers or names of the drones to check the rules of the country he wants to fly in. That way he can find out which are good to go and which are mostly illegal.

5.2.3 Flybot

On January 29th, 2015 we spoke to the drone operator at Flybot. Their preparations before take-off when flying drones are especially looking at the weather, solar activity and an on-site check for high structures. The weather is checked more specifically for wind speed and direction. The Flybot drone can handle some wind up to five Beaufort but they stop flying immediately if the wind is stronger than that. Solar activity is checked with a mobile application called Solar Activity Monitor. Especially the k-index is checked. The k-index gives a value between zero and nine which represents the force of a geomagnetic storm on earth caused by the sun. The drone operator only flies when the k-index is below four. According to him it would be a useful extension to the application. On location he especially looked out for high objects. It is very difficult to see how high objects are and whether the drone is just above or just below the top of an object.

Furthermore he told us that according to his experience it is very difficult to fly a drone legally correct in Belgium clearance must be asked for at the DVLG. The distance between a runway and the drone should be three nautical miles which is more than just five km. In Belgium hobbyists are only allowed to fly on a hobbyist field or if younger than 14 and they don't fly higher than their house in their own garden, with a toy drone. He recommended to include nuclear plants and NOTAMS.

5.3 Google Analytics Data

RuleMaDrone was released on May the 21st of 2015. Google Analytics was used to collect usage data on the website. In this section we will discuss the collected data. At the time of writing the website had been online for 22 days.

During the period that the website was online we discovered that there is a good amount of interest in this kind of an application. RuleMaDrone had an average amount of forty users a day, this resulted in a total of 877 users over the period of 22 days. On the third of June only, there were 118 visitors on the website. The users visiting RuleMaDrone came from sixty different countries spread all over the world, as can be seen in Fig. 5.7. The top locations for visiting our website were the United States, Belgium, China and Germany. This shows that there is a global interest in such an application.

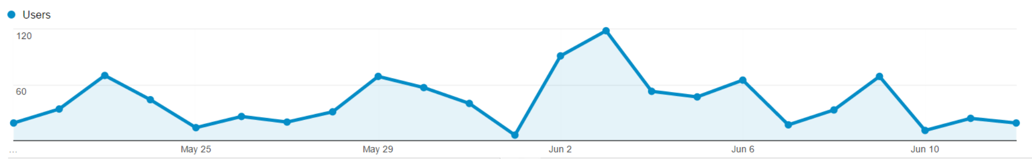


Figure 5.6: Amount of visitors per day.

Most users (601) visited our website with a direct link to the site. Other users were reached with social media: 170 with Facebook, 17 with Google+ and 2 via LinkedIn. 81 users found us with Google search. An overview from how different sessions were initiated can be found in

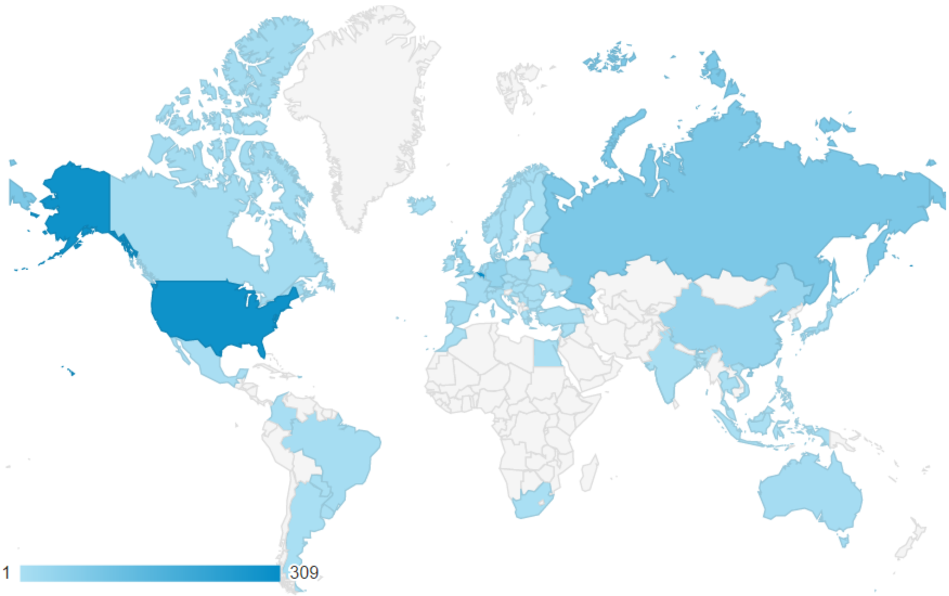


Figure 5.7: Geographic spreading of users

Fig. 5.8.

1.	Direct	676
2.	Social	252
3.	Referral	101
4.	Organic Search	81

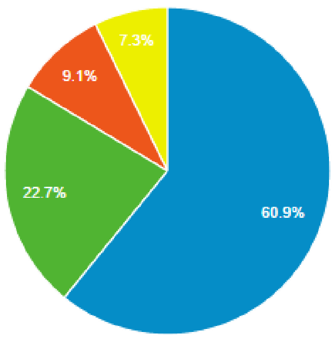


Figure 5.8: How did users find RuleMaDrone

In this rather short period of time one person added the rules for the Netherlands. Besides this person, no users added any new rules to the system but we believe that was to be expected when only just launching an website with a crowdsourcing system.

We also received some positive reactions such as *I'm a big fan of the* *page. It's just what people need with all these different rules!* and *it is going to be useful and easy to use for future drone pilots.*

Chapter 6

Conclusion

In this thesis we presented RuleMaDrone. RuleMaDrone is the first online mapping application which includes all three factors of our drone safety framework. This makes RuleMaDrone a very complete experience for drone operators to use in comparison to other tools. RuleMaDrone also integrates a crowdsourcing component. It provides users with the option of adding rules themselves with a natural language interface and so helping other people with finding a safe fly zone. Helping other people is one of the best motivators to achieve high quality in crowdsourcing (12) (13).

The possibility for the user to choose the location of flight and calculate a fly zone at that location is unique within comparable tools. This is also the feature that intertwines the three pillars of drone safety. Forbidden areas are removed from the fly zone. The user is warned when the wind is to strong to fly that type of drone or when there is geomagnetic storm which can cause problems with GPS flight.

Providing an option to integrate the rules from all over the world is unique for current available tools. That is why we chose for a natural language interface to crowdsource different rules. Everyone is able to add a rule via that interface. All we have to do is verify the rule, once we have done that the rule is automatically added to the map thanks to our back-end.

The current limitations of RuleMaDrone are researched in other work.

First of all we are still dependent on OpenStreetMap for the correctness of our rule layers. If the OpenStreetMap data is incorrect or incomplete our layers are too. The accuracy and completeness of OpenStreetMap is discussed in (14) and (15). The second limitation is the current state of crowdsourcing geographic information. Research showed that more than ninety percent of the users only consume data but they do not contribute (16) (17). Also on RuleMaDrone a small amount of users contributed thus far. As soon as more rules are added the user collaboration will rise as shown by Kittur et al. (18).

Our studies show that the need for such a tool was urgent. Despite the fact that most people are not negative regarding the rise of consumer and commercial drones, they do have some concerns. Almost nobody knows the current laws and regulations. Professional drone operators helped us in our implementation process to create RuleMaDrone.

6.1 Extending to other Space Usage Rules

RuleMaDrone is developed to be easy extensible to other space usage rules. To do this, for example, for space usage rules such as *alcohol consumption*, *no smoking* or *no fishing*, only a document has to be added to the collection *tags*. The document has two fields: tag and text. Adding a document as can be seen in 2 makes an new URL available. The URL of the map for drone flying can be for example `www.rulemadrone.org/map?country=BE&lat=50.8858391&lng=5.534101&tag=drone_flying` which shows the rules regarding drone flying in Belgium and places the marker on the according coordinates. With the *tag* document added to the database one can change the parameter in the according tag and the map for that space usage will appear. In our example the URL will look like this: `www.rulemadrone.org/map?country=BE&lat=50.8858391&lng=5.534101&tag=alcohol_consumption`. By adding rules to the database, one could get a map as in Fig. 6.1.

Code 2: The collection tags in MongoDB.

```
[
  { "tag":"drone_flying",
    "text": fly your drone
  },
  {
    "tag":"alcohol_consumption",
    "text":consume_alcohol
  }
]
```

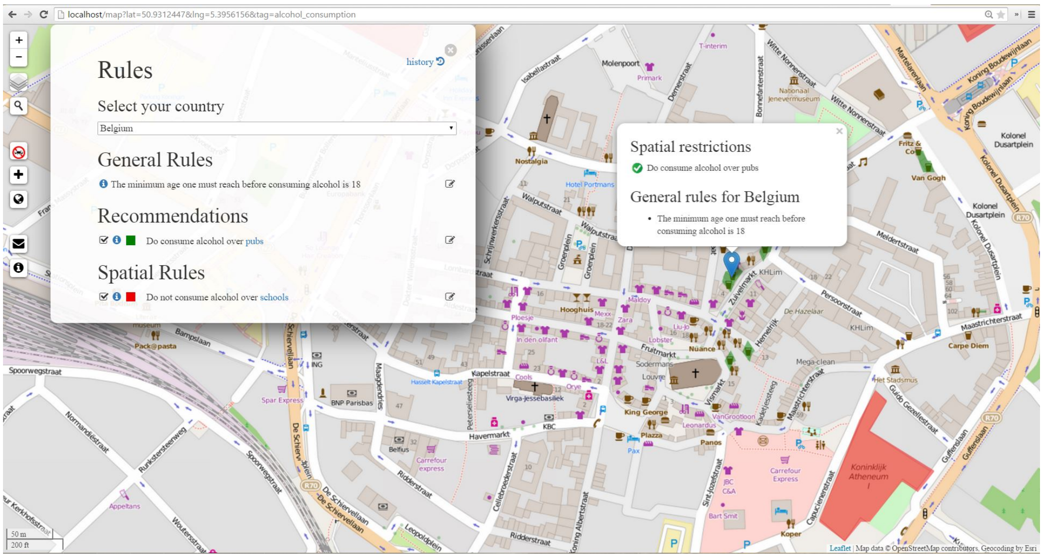


Figure 6.1: RuleMaDrone for the space usage rules regarding alcohol consumption

The only feature that will not yet be available are the general rules. To make adding and editing general rules available one has to insert the general rules that must be possible into the database. An example of how this is done can be found in code 3.

Code 3: A document in the collection general rules. This document includes the parameter that defines the rule and the HTML code which will be displayed when one wants to add a general rule.

```
{
        "ruletag" : "alcohol_consumption",
        "parameter" : "minimum age",
        "html" : "The minimum age one must reach before
           consuming alcohol is <input name='min_age'
           required> </input>"
}
```

Chapter 7

Future Work

The work started in this thesis can be proceeded in four different directions thanks to the flexible pipeline of RuleMaDrone, see Fig. 4.1.

7.1 Introducing a Fourth Pillar

In this thesis we focussed on the three pillars in the framework presented in Fig. 1.5. It could be argued that there is a need for a fourth pillar in this framework: the drone operator. To show how the drone operator would relate to the other components we again present two questions and link them to the new drone safety framework in Fig. 7.1.

1. Am I licensed to fly this type of drone?
2. Am I licensed to fly at night?

7.2 Integration With Other Platforms

The pipeline of RuleMaDrone enables developers to write other applications. Drone operators are asking for a mobile version of RuleMaDrone. Since all layers are divided in different segment, it is even possible to develop an application which is very economical with data. When a user for example asks for a fly zone, only the segments of the rule layer close to the fly zone will be sent to prevent using too much mobile data.

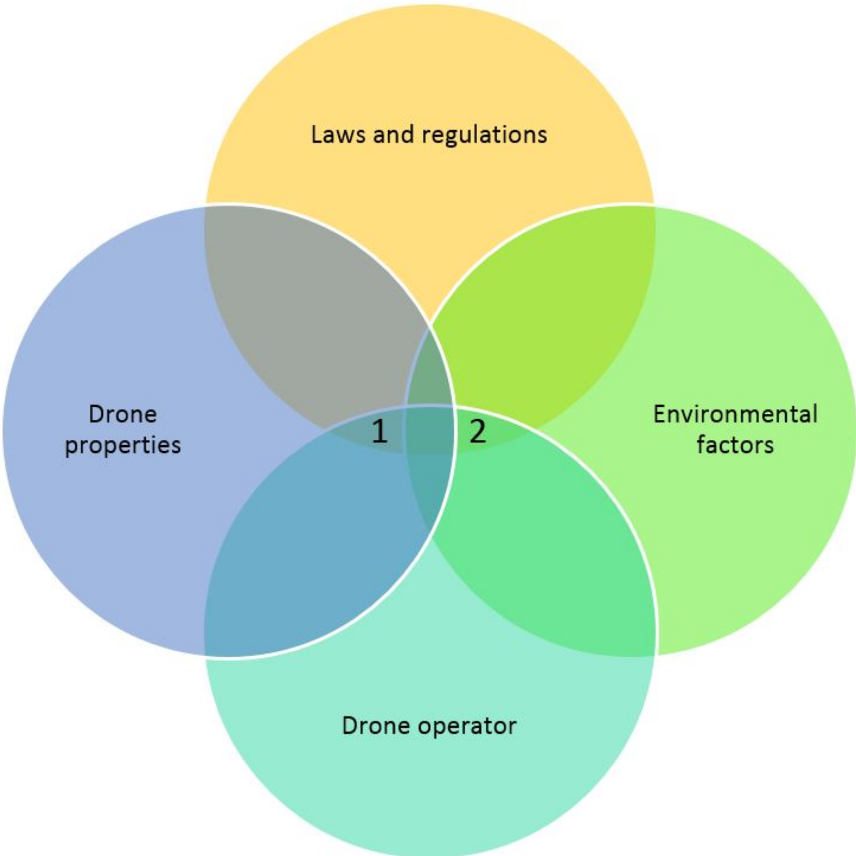


Figure 7.1: The drone safety framework including the drone operator component

A virtual reality application integrated in Google Earth would be helpful for a user to check out the flying environment from his home. This could be useful to check on high objects like buildings and/or trees before actually going on location.

7.3 From Planning Tool to Live Working Tool

Several drones already provide the option to control it while looking through the camera mounted on the drone itself. It would be very helpful to use the data provided by RuleMaDrone and integrate it in a virtual reality application. A drone pilot would then be able to immediately view when the drone comes close to an area where it is forbidden to fly. When flying a drone without virtual reality glasses, the integration of the RuleMaDrone data in the controller would be useful for drone operators. A drone controlled by a smartphone could, for example, warn a user when he is flying into restricted area's.

7.4 Rule Collection

As can be seen in this thesis rule information is collected with semi natural language. More research can be done to convert actual natural language into space usage rules. This could also extend the range of space usage rules that could be provided by the tool. Space usage rules are now always defined for the whole country while it could be possible that some rules only apply to a city or a certain part of the country.

Bibliography

[1] I. Hong, M. Kuby, and A. Murray. Deviation flow refueling location model for continuous space: commercial drone delivery system for urban area. 2015.

[2] B. Elzweig. Civilian commercial drones are coming; are we ready? *Southern Law Journal*, 25(1):161, 2015.

[3] C. Roos. Human error in operating mini rpas. 2014.

[4] R. Clarke and L.B. Moses. The regulation of civilian drones' impacts on public safety. *Computer Law & Security Review*, 30(3): 263–285, 2014.

[5] E. Carr. Unmanned aerial vehicles: examining the safety, security, privacy and regulatory. 2014.

[6] R. Clarke. Understanding the drone epidemic. *Computer Law & Security Review*, 30(3):230–246, 2014.

[7] J. Schöning, B. Hecht, and W. Kuhn. Informing online and mobile map design with the collective wisdom of cartographers. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 765–774. ACM, 2014.

[8] P. Samsonov, X. Tang, J. Schöning, W. Kuhn, and B. Hecht. You can't smoke here: Towards support for space usage rules in location-aware technologies. 2014.

[9] M. Poblet. Visualizing the law: crisis mapping as an open tool for legal practice. *J. Open Access L.*, 1:1, 2013.

[10] R. Hoekstra, R. Winkels, and E. Hupkes. Reasoning with spatial plans on the semantic web. 2009.

[11] R. Rao, S.G Krishna, J.V. Prasad, S. Prasad, D. Prasad, and K. Niranjan. Geomagnetic storm effects on gps based navigation. *Ann. Geophys*, 27(5):2101–2110, 2009.

[12] J. Rogstadius, V. Kostakos, A. Kittur, B. Smus, J. Laredo, and M. Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *ICWSM*, 2011.

[13] A. Finnerty, P. Kucherbaev, S. Tranquillini, and G. Convertino. Keep it simple: Reward and task design in crowdsourcing. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, page 14. ACM, 2013.

[14] C. Keßler, R. de Groot, and A. Theodore. Trust as a proxy measure for the quality of volunteered geographic information in the case of openstreetmap. In *Geographic information science at the heart of Europe*, pages 21–37. Springer, 2013.

[15] M. Haklay. How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets. *Environment and Planning B Planning and Design*, (37): 682–703, 2010.

[16] L. Guo, E. Tan, S. Chen, X. Zhang, and Y.E. Zhao. Analyzing patterns of user content generation in online social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 369–378. ACM, 2009.

[17] P. Neis and D. Zielstra. Recent developments and future trends in volunteered geographic information research: The case of openstreetmap. *Future Internet*, 6(1):76–106, 2014.

[18] A. Kittur, E. Chi, B. Pendleton, B. Suh, and T. Mytkowicz. Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. *World wide web*, 1(2):19, 2007.